

Código Cliente

```
1 import java.rmi.registry.LocateRegistry;
2 import java.rmi.registry.Registry;
3
4 public class Cliente {
5     private Cliente() {}
6     public static void main(String[] args) {
7         String host = (args.length < 1) ? null : args[0];
8         try {
9             Registry registry = LocateRegistry.getRegistry(host);
10            //también puedes usar getRegistry(String host, int port)
11            //Calculadora stub = (Calculadora) registry.lookup("Suma");
12            Calculadora stub = (Calculadora) registry.lookup("Resta");
13            //Calculadora stub= (Calculadora) registry.lookup("Multiplicacion");
14            //Calculadora stub = (Calculadora) registry.lookup("Division");*/
15
16            int x=9,y=3;
17            int response = stub.suma(x,y);
18            System.out.println("respuesta de sumar "+x+" y "+y+" : " + response);
19            int response2 = stub.resta(x,y);
20            System.out.println("respuesta de restar "+x+" y "+y+" : " + response2);
21            int response3 = stub.multi(x,y);
22            System.out.println("respuesta de multiplicar "+x+" y "+y+" : " + response3);
23            double response4 = stub.div(x,y);
24            System.out.println("respuesta de dividir "+x+" y "+y+" : " + response4);
25
26        } catch (Exception e) {
27            System.err.println("Excepción del cliente: " + e.toString());
28            e.printStackTrace();
29        }
30    }
31 }
```

Código Calculadora.java

```
1 import java.rmi.Remote;
2 import java.rmi.RemoteException;
3
4 public interface Calculadora extends Remote {
5     int suma(int a,int b) throws RemoteException;
6     int resta (int a,int b) throws RemoteException;
7     int multi(int a,int b) throws RemoteException;
8     double div (int a,int b) throws RemoteException;
9
10 }
```

Código Servidor

```
1 import java.rmi.registry.Registry;
2 import java.rmi.registry.LocateRegistry;
3 import java.rmi.RemoteException;
4 import java.rmi.server.UnicastRemoteObject;
5 //import Suma.*;
6
7 public class Servidor implements Calculadora{
8
9     public Servidor() {}
10
11     public int suma(int a, int b) {
12         return a+b;
13     }
14
15     public int resta(int a,int b){
16         return a-b;
17     }
18
19     public int multi(int a, int b){
20         return a*b;
21     }
22
23     public double div(int a, int b){
24         return a/b;
25     }
26
27     public static void main(String args[]) {
28         try {
29             //puerto default del rmiregistry
30             java.rmi.registry.LocateRegistry.createRegistry(1099);
31             System.out.println("RMI registro listo.");
32         } catch (Exception e) {
33             System.out.println("Excepcion RMI del registry:");
34             e.printStackTrace();
35         } //catch
36         try {
37             System.setProperty("java.rmi.server.codebase","file:/c:/Temp/Calculadora/");
38             Servidor obj = new Servidor();
39             Calculadora stub = (Calculadora) UnicastRemoteObject.exportObject(obj, 0);
40             // Ligamos el objeto remoto en el registro
41             Registry registry = LocateRegistry.getRegistry();
42             registry.bind("Suma", stub);
43             registry.bind("Resta", stub);
44             registry.bind("Multiplicacion", stub);
45             registry.bind("División", stub);
46
47             System.err.println("Servidor listo...");
48         } catch (Exception e) {
49             System.err.println("Excepción del servidor: " + e.toString());
50             e.printStackTrace();
51         }
52     }
53 }
54 }
```