



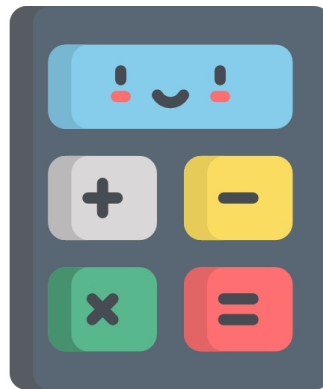
Instituto Politécnico Nacional

Escuela Superior de Cómputo

---

UA.Aplicaciones para Comunicaciones  
en Red

“Tarea 10.1 : Calculadora RMI”



**Alumno:**

Márquez León Jorge Luis

**Profesor:**

Ricardo Martínez Rosales

**Grupo:**

3CM16

## Interface Calculadora.java

---

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface Calculadora extends Remote {

    int suma(int a, int b) throws RemoteException;

    int resta(int a, int b) throws RemoteException;

    int multiplicacion(int a, int b) throws RemoteException;

    float division(float a, float b) throws RemoteException;

}
```

## Servidor.java

---

```
import java.rmi.registry.Registry;
import java.rmi.registry.LocateRegistry;
import java.rmi.server.UnicastRemoteObject;
//import Suma.*;
public class Servidor implements Calculadora {

    public Servidor() {
    }

    public int suma(int a, int b) {
        System.out.println("Estoy en el servidor RMI y soy una suma.");
        return a + b;
    }

    public int resta(int a, int b) {
        System.out.println("Estoy en el servidor RMI y soy una resta.");
        return a - b;
    }

    public int multiplicacion(int a, int b) {
        System.out.println("Estoy en el servidor RMI y soy una multiplicacion.");
        return a * b;
    }

    public float division(float a, float b) {
        System.out.println("Estoy en el servidor RMI y soy una division.");
        return a / b;
    }

    public static void main(String args[]) {
        try {
            //puerto default del rmiregistry
```

```

    java.rmi.registry.LocateRegistry.createRegistry(1099);
    System.out.println("RMI registro listo.");
} catch (Exception e) {
    System.out.println("Excepcion RMI del registry:");
    e.printStackTrace();
} //catch
try {
    //System.setProperty("java.rmi.server.codebase", "file:/c:/Temp/Suma/");
    Servidor obj = new Servidor();
    Calculadora stub = (Calculadora) UnicastRemoteObject.exportObject(obj, 0);
    // Ligamos el objeto remoto en el registro
    Registry registry = LocateRegistry.getRegistry();
    registry.bind("Calculadora", stub);

    System.err.println("Servidor listo...");
} catch (Exception e) {
    System.err.println("Excepcion del servidor: " + e.toString());
    e.printStackTrace();
}
}
}

```

## Cliente.java

---

```
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.util.Scanner;

public class Cliente {

    private Cliente() {
    }

    public static void main(String[] args) {
        try {
            boolean salir = false;
            while (!salir) {
                Registry registry = LocateRegistry.getRegistry("localhost", 1099);
                //también puedes usar getRegistry(String host, int port)
                Calculadora stub = (Calculadora) registry.lookup("Calculadora");
                Scanner leer = new Scanner(System.in);

                System.out.println("\nCalculadora");

                System.out.println("1.Suma");
                System.out.println("2.Resta");
                System.out.println("3.Multiplicacion");
                System.out.println("4.Division");
                System.out.println("5.Salir");
                System.out.println("Elige una opcion:");
                int elegir = leer.nextInt();

                int response, x, y;
                switch (elegir) {
                    case 1:
```

```

System.out.println("Primer numero:");
x = leer.nextInt();
System.out.println("Segundo numero:");
y = leer.nextInt();

response = stub.suma(x, y);
System.out.println("Respuesta Sumar: " + x + " + " + y + " = " +
response);
    break;
case 2:
    System.out.println("Primer numero:");
    x = leer.nextInt();
    System.out.println("Segundo numero:");
    y = leer.nextInt();

    response = stub.resta(x, y);
    System.out.println("Respuesta Restar: " + x + " - " + y + " = " +
response);
    break;
case 3:
    System.out.println("Primer numero:");
    x = leer.nextInt();
    System.out.println("Segundo numero:");
    y = leer.nextInt();

    response = stub.multiplicacion(x, y);
    System.out.println("Respuesta Multiplicar: " + x + " * " + y + " = " +
response);
    break;
case 4:
    System.out.println("Primer numero:");
    x = leer.nextInt();
    System.out.println("Segundo numero:");
    y = leer.nextInt();

```

```

        float response_div = stub.division(x, y);
        System.out.println("Respuesta Dividir: " + x + " / " + y + " = " +
response_div);
        break;
    case 5:
        salir = true;
        break;
    default:
        System.out.println("Opcion no valida");
        break;
    }

}

} catch (NotBoundException | RemoteException e) {
    System.err.println("Excepcion del cliente: " + e.toString());
}
}}

```