

Bloque 6. Paradigmas de la Programación

Tarea 1: Aplicaciones con RMI

Enunciado

1.- Crear una aplicación distribuida con RMI, que permita realizar las siguientes operaciones:

- Dados dos números, que devuelva como resultado la multiplicación de estos. Por ejemplo: multiplica(2,3)=6
- Dados dos números, que devuelva como resultado la potencia del primero con respecto al segundo. Por ejemplo: potencia(2,3)=8

A tener en cuenta:

- La clase Interface contiene el método remoto.
- La clase que implementa la interface remota será el objeto remoto. Contiene el código que realizará la operación de multiplicación o potencia.
- La clase Servidor será un Main y registrará el objeto remoto.
- La clase Cliente será un Main y buscará el objeto remoto. En esta clase el usuario podrá elegir entre la opción de multiplicar o hallar la potencia de un número.

Código Interface remota:

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

public interface InterfaceOperaciones extends Remote {
    int multiplicacion(int num1, int num2, int solucion) throws RemoteException; //Método que se publica
    int potencia(int num1, int num2, int solucion) throws RemoteException;
}
```

Código Clase que implementa la interface remota:

```
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class Operaciones extends UnicastRemoteObject implements InterfaceOperaciones{
    public Operaciones() throws RemoteException {}//Constructor vacío

    public int multiplicacion(int num1, int num2, int solucion) throws RemoteException {// Implementación del
método remoto
        solucion = (num1 * num2);
        return solucion;
    }
    public int potencia(int num1, int num2, int solucion) throws RemoteException {// Implementación del
método remoto
        solucion = (int) Math.pow(num1, num2);
        return solucion;
    }
}
```

Código Clase Servidor:

```
import java.rmi.Naming;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class Servidor {

    public static void main(String[] args) {
        try {
            //Crea instancia del objeto que implementa la interfaz (objeto a registrar):
            System.out.println("Servidor Arrancado....");

            Operaciones obj = new Operaciones();
            Registry registry = LocateRegistry.createRegistry(1099); //Arranca rmiregistry local en el puerto
1099 Naming.rebind("//127.0.0.1/ObjetoOperaciones", obj); //Hace visible el objeto para clientes
            System.out.println("El Objeto Operaciones ha quedado registrado");

        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
            e.printStackTrace();
        }
    }
}
```

Código Clase Cliente:

```
import static java.lang.System.exit;
import java.rmi.Naming;
import java.util.Scanner;

public class Cliente {

    public static void main(String[] args) {

        int num1, num2, resultado;
        resultado = 0;
        int opcion;
        Scanner sc = new Scanner(System.in);
        try {
            System.out.println("*****CALCULADORA*****");
            System.out.println("Introduce un primer valor: ");
            num1 = sc.nextInt();
            System.out.println("Introduce un segundo valor: ");
            num2 = sc.nextInt();

            System.out.println("¿Qué operación deseas realizar? 1-> Multiplicacion 2-> Potencia");
            opcion = sc.nextInt();
        }
    }
}
```

```

//Localiza el objeto distribuido:
InterfaceOperaciones obj = (InterfaceOperaciones)
Naming.lookup("//127.0.0.1/ObjetoOperaciones");

if (opcion == 1) {
    System.out.println("Opción de multiplicación");
    resultado = obj.multiplicacion(num1, num2, resultado);
} else if (opcion == 2) {
    System.out.println("Opción de potencia");
    resultado = obj.potencia(num1, num2, resultado);
} else {
    System.out.println("Opción no válida");
    exit(opcion);
}

System.out.println("Resultado de la operacion " + resultado);
} catch (Exception e) {
    System.out.println("Excepción : " + e.getMessage());
    e.printStackTrace();
}
}
}

```

Resultado:

Desde Cliente...	Desde Servidor...
run: *****CALCULADORA***** Introduce un primer valor: 3 Introduce un segundo valor: 7 ¿Qué operación deseas realizar? 1-> Multiplicacion 2-> Potencia 1 Opción de multiplicación Resultado de la operacion 21 BUILD SUCCESSFUL (total time: 6 seconds)	run: Servidor Arrancado.... El Objeto Operaciones ha quedado registrado BUILD STOPPED (total time: 12 seconds)