

Bloque 6. Paradigmas de la Programación

Tarea 3: Comprobar si es primo con RMI

Enunciado

3.- Crear una aplicación distribuida con RMI, a la que se le envíe un número y nos devuelva como resultado si el número es primo o no.

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 evaluará si un número es primo o no.
- La clase Servidor será un Main y registrará el objeto remoto.
- La clase Cliente será un Main y buscará el objeto remoto

Código Interface remota:

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

public interface InterfacePrimo extends Remote{
    int compruebaPrimo(int numero, int esPrimo) throws RemoteException; //Método que se publica
}
```

Código Clase que implementa la interface remota:

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

public class CompruebaPrimo extends UnicastRemoteObject implements InterfacePrimo {
    public CompruebaPrimo() throws RemoteException {} //Constructor vacío

    public int compruebaPrimo(int numero, int esPrimo) throws RemoteException { // Implementación del método remoto
        if (numero <= 1) { // Si es 1 o menor, directamente no será primo
            esPrimo = 0;
        } else { // Si es mayor, realizo la comprobación
            for (int i = 2; i <= numero / 2; i++) {
                if ((numero % i) == 0) {
                    esPrimo = 0;
                } else {
                    esPrimo = 1;
                }
            }
        }
        return esPrimo;
    }
}
```

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....");

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

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

Código Clase Cliente:

```
import java.rmi.Naming;
import java.util.Scanner;

public class Cliente {

    public static void main(String[] args) {
        int numero, esPrimo;
        esPrimo = 0;
        String resp;
        Scanner sc = new Scanner(System.in);

        try {
            System.out.println("***Introduce un número entero: ** ");
            numero = sc.nextInt();
            System.out.println("Vamos a comprobar si " + numero+ " es primo");

            //Localiza el objeto distribuido:
            InterfacePrimo obj = (InterfacePrimo) Naming.lookup("//127.0.0.1/ObjetoPrimo");
            esPrimo = obj.compruebaPrimo(numero, esPrimo); //Llama al método calcularEdad
        }
    }
}
```

```

    if (esPrimo == 1){
        resp = " Es PRIMO";
    }else{
        resp = " NO es PRIMO";
    }
    System.out.println("El número " + numero + resp);

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

```

Resultado:

Desde Cliente...	Desde Servidor...
run: **Introduce un número entero: ** 83 Vamos a comprobar si 83 es primo El número 83 Es PRIMO BUILD SUCCESSFUL (total time: 6 seconds)	run: Servidor Arrancado.... El Objeto CompruebaPrimo ha quedado registrado BUILD STOPPED (total time: 16 seconds)