ejemplo

### Proyecto 001-Ejercicio Inicial

**Enunciado del programa (Como los enunciados de clase (toma de los ejemplos))**

**La arquitectura del programa**

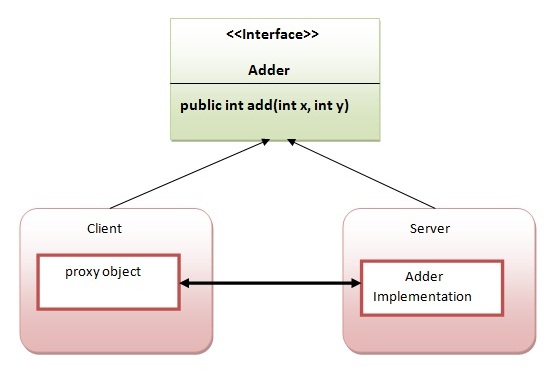
**- Explicar detalles que no aparecen a simple vista en el codigo, Interfaces empleadas, que hace cada una, porque se organiza como se organiza**

**- Explicar la arquitectura, que parte va en el servidor, y en el cliente , y cual es la interfaz en comun si es que la tienen. Y como se comunican entre ellos (que informacion se mandan)**

NO PONER EL PANTALLAZO DEL WORKSPACE ES PREFERIBLE EL ESQUE MA DE LAS CLASES

Enunciado

#### Explicacion

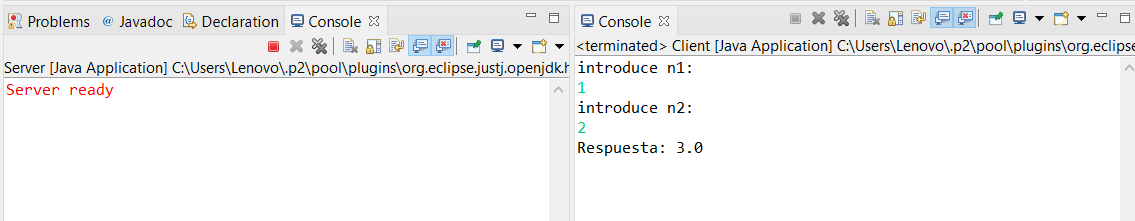
En este ejemplo, la aplicación cliente sólo necesita dos archivos, la interfaz remota y la aplicación cliente. En la aplicación rmi, tanto el cliente como el servidor interactúan con la interfaz remota. La aplicación cliente invoca métodos en el objeto proxy, RMI envía la petición a la JVM remota. El valor de retorno es enviado de vuelta al objeto proxy y luego a la aplicación cliente. 

### Para crear la interfaz remota, extienda la interfaz Remote y declare la RemoteException con todos los métodos de la interfaz remota. Aquí, estamos creando una interfaz remota que extiende la interfaz Remote. Sólo hay un método llamado add() y declara RemoteException.

Para crear la interfaz remota, extienda la interfaz Remote y declare la RemoteException con todo Ahora proporcione la implementación de la interfaz remota. Para proporcionar la implementación de la interfaz Remote, necesitamos extender la clase UnicastRemoteObject,o utilizar el método exportObject() de la clase UnicastRemoteObject. En caso de que extiendas la clase UnicastRemoteObject, debes definir un constructor que declare RemoteException.

En el cliente estamos obteniendo el objeto stub mediante el método lookup() del servidor de Naming e invocando el método sobre este objeto.

En este ejemplo, estamos ejecutando las aplicaciones servidor y cliente, en la misma máquina utilizando el puerto 1099 de LocateRegistry.

Output (explicacion de como es la solucion)

EJERCICIOS

#### 001- Hello world con naming (modulo antiguo)

**Enunciado**

Java Rmi es una herramienta que tiene sus años y ha pasado por 2 etapas principales en su vida util, para aprender de sus origenes, ejecuta el programa, y entiende su funcionamiento. (no tiene que funcionar tranquilos)

**Explicacion**

Como el codigo se intenta ejecutar sin haber realizado la compilacion del codigo ni el levantamiento del servicio Rmi, eclipse no sera capaz de ejecutar el codigo, y fallara, ya que no sera capaz de establecer la conexion.

**Output**



#### 002 - Compilacion y ejecucion (modulo antiguo)

**Enunciado**

Haz funcionar el ejemplo anterior haciendo todo el proceso de complacion. Las instrucciones son las siguientes:

1. Habre el powershell y colocate en la carpeta src del proyecto
2. Ejecuta el comando **javac <nombre del archivo java>** , y azlo con cada archivo .java del src
3. Crea los objetos Stub y skeleton con **rmic AdderRemote**
4. Ejecuta el comando **rmiregistry 5000** , para ejecutar el servidor rmi en el puerto 5000
5. Sin cerrar el terminal anterior abre un nuevo terminal, en el src y ejecuta **java MyServer** , para empezar el servidor.
6. En otra terminal nueva con **java MyClient** empieza el cliente.

**Explicacion**

Con los comandos mencionados anteriormente hacemos lo siguiente:

* Javac <archivo .java> -> compilamos el codigo fuente.
* rmic AdderRemote   -> creamos los objetos stub y skeleton utilizando la herramienta rmic .
* rmiregistry 5000  -> empezamos el servicio rmi en el puerto indicado (por defecto es 5000),
* java MyServer  -> empezamos la ejecucion del servidor
* java MyClient  -> empezamos la ejecucion del cliente

Como curiosidad, si levantamos este servicio rmi en la linea de comandos, la ejecucion del servidor y el cliente se pueden hacer por ejecucion normal en eclipse. Debido a que el servicio permite la interconectividad entre servidor y cliente a traves del puerto 5000.

**Output**

Terminal 1 el de la compilacion y levantamiento del servicio



-> El metodo que estamos utilizando esta deprecated y el sistema nos lo notifica.

Terminal 2: el server



Terminal 3: el cliente



-> Ejecucion alterna en eclipse si el servicio rmi esta levantado



003- Hello world con Registry.

**Enunciado**

RMI despues de mucho tiempo se ha actualizado y elementos que antaño habian sido un pilar fundamental de la plataforma han sido sustituidos por otros mas faciles de usar. Actualiza el programa de hello world con la interfaz Registry.

**Explicacion**

004- Ejercicio de suma

**Enunciado**

**Explicacion**

**Output**

005- Calculadora

**Enunciado**

Se quiere explandir la capacidad de calculo del programa anterior, expandiendo los tipos de calculo que es capaz de hacer. Modifica el programa, para que pueda realizar las siguientes acciones:

1. Sumar
2. Restar
3. Multiplicar
4. Dividir
5. Terminar el programa.

**Explicacion**

**Output**