ejemplo

### Proyecto 001-Ejercicio Inicial

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

**La arquitectura del programa**

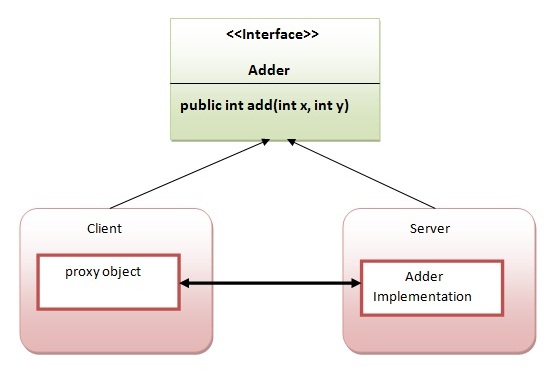
**- Explicar detalles que no aparecen a simple vista en el codigo, Interfaces empleadas, que hace cada una, porque se organiza como se organiza**

**- 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

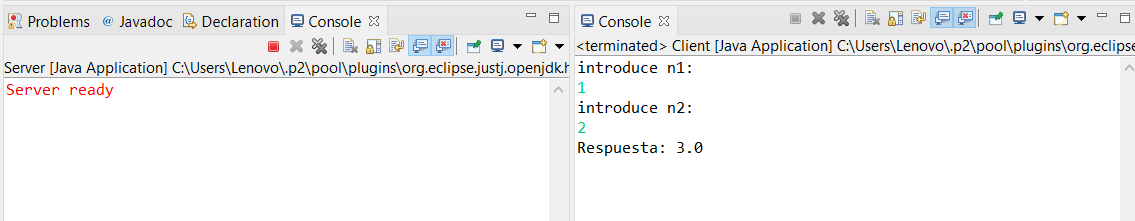
En este ejemplo, la aplicación cliente sólo necesita dos archivos, la interfaz remota y la aplicación cliente. En la aplicación rmi, tanto el cliente como el servidor interactúan con la interfaz remota. La aplicación cliente invoca métodos en el objeto proxy, RMI envía la petición a la JVM remota. El 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 namig

**Enunciado**

Ejecuta el programa, y entiende su funcionamiento.

**Explicacion**

**Output**

002 - Compilacion y ejecucion del modulo (antiguo)

**Enunciado**

**Explicacion**

**Output**

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**