



Licenciatura en Ingeniería de Software  
Taller de Programación Visual III

UNIDAD IV Programación Distribuida y Paralela	Tarea # 9
Nombre: Luis Enrique Santos Landeros	Fecha:13/01/19

## JAVA RMI

### ¿Qué es RMI?

Antes de responder esta fundamental pregunta, debemos remontarnos a la base misma de rmi, vale decir, lo que es conocido como Remote Procedure Calling (RPC)

### ¿Qué es RPC?

Una definición formal de RPC seria:

RPC es la transferencia sincrónica de datos y control entre dos partes de un programa distribuido a través de espacios de direcciones disjuntas.

el llamado remoto de procesamiento es, procesar de manera remota algún segmento de código (utilizando una CPU, Memoria, Stack, etc, remotos), necesario localmente, sin salir del procesamiento local.

Este tema es bastante más complicado de lo que parece, ya que hay que tener en cuenta una serie de detalles y problemas, como, por ejemplo:

Marshalling y Unmarshalling: Estandarizar el orden de los datos

Inestabilidad de la Red: Como retornar a la programación

Seguridad: No cualquiera pueda interrumpir mi flujo de procesamiento.

Representación de Datos: Por Ejemplo, UTF8 v/s ASCII

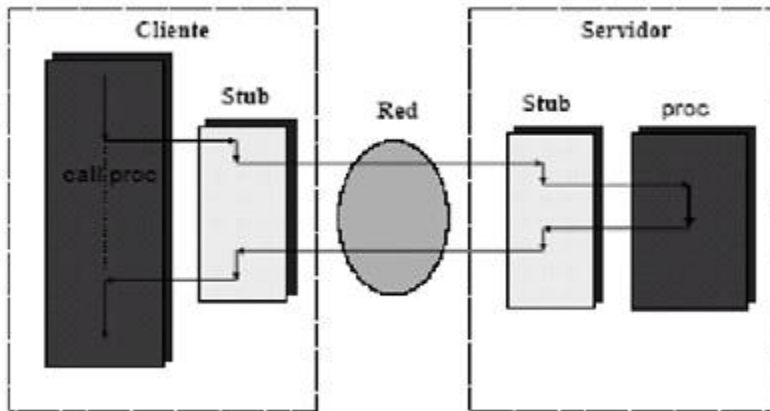
Distribución de Hardware: Distinta zona de memoria, stack, clock de la CPU, etc

Estos problemas, obviamente, son aplicables también a RMI.

Ahora conociendo estos conceptos podemos saber más claramente que el RMI.

RMI es un paquete de JAVA que permite manejar objetos (y sus respectivos métodos) de manera remota, para utilizar los recursos de un servidor de manera transparente para el usuario local.

La manera en que RMI (y RPC en general) logra hacer esto, es por medio de lo que se conoce como STUBs. En el caso del STUB servidor, se conoce como SKELETON. Estos Stubs y Skeletons permiten que al momento de ser invocada la función remota esta pueda ser "simulada localmente"



Para la comunicación entre el servidor y el cliente, se trabaja con interfaces, que deben ser implementadas por el servidor y/o cliente, para que los STUBs puedan realizar la transparencia para ambos. Además, esto evita que deba existir una definición local real de la clase remota, vale decir, en el cliente solo debe estar definida la interface, no la clase remota.

Otro punto importante en RMI, es el cómo se produce la conectividad entre el cliente y servidor. Para esto se ocupa una herramienta de JAVA, llamada RMI Registry.

El RMI Registry puede estar localizado en un lugar distinto al servidor, y se encarga de registrar un determinado objeto y asignarle un servidor que se encargará de procesar dicho objeto.

El funcionamiento general es:

Se ejecuta el RMI Registry, en algún lugar de la red.

El servidor que desea manejar un objeto, se registra en dicho servidor,

El RMI Registry registra el par: OBJETO/SERVIDOR

El cliente que necesita utilizar un determinado objeto, hace una consulta al RMI Registry, quien devuelve el STUB listo para la comunicación

Esto es, a grandes rasgos, lo más indispensable que se debe saber a la hora de utilizar RMI. Se puede profundizar mucho más sobre el tema, en la página oficial de JAVA, en donde se pueden encontrar más ejemplos sobre el tema, en la API se encuentran todos los métodos y clases del paquete RMI, y en Internet.



## Windows Communication Foundation (WCF)

### ¿Qué significa Windows Communication Foundation (WCF) ?

Windows Communication Foundation (WCF) es una tecnología para desarrollar aplicaciones basadas en arquitectura orientada a servicios (SOA). WCF se implementa utilizando un conjunto de clases ubicadas en la parte superior del Common Language Runtime (CLR) .NET. Aborda el problema de la interoperabilidad utilizando .NET para aplicaciones distribuidas.

Las colaboraciones interempresariales se han vuelto cada vez más comunes, y diferentes organizaciones comerciales utilizan software diferente como su implementación subyacente. Por lo tanto, se requiere un marco de comunicación independiente de la plataforma y del lenguaje de programación para facilitar la comunicación.

**Windows Communication Foundation**, originalmente conocida por el nombre de Índigo, es la parte encargada, como su propio nombre indica, de las comunicaciones y mensajería, aunque su principal uso son los servicios web.

Hasta el momento, el servicio de mensajería entre aplicaciones se realizaba mediante los protocolos COM, DCOM o MSQM, que obligaba a los programadores a ceñirse no sólo a una forma de programación concreta, sino que también estaba atada a la plataforma y al lenguaje de programación

### Tipos de servicios web

A día de hoy existe una gran variedad de protocolos sobre los que los servicios web pueden operar, pero son dos los protocolos estrella que la práctica totalidad de los servicios web utilizan hoy día:

**SOAP:** Simple Object Access Protocol. Creado en 1998, se sirve de mensajes XML para el intercambio de mensajes. Puede operar sobre cualquier protocolo de transporte, aunque lo más común es que lo haga sobre HTTP o HTTPS. Es el protocolo más común en servicios web de carácter privado.

**REST:** REpresentational State Transfer. Concepto surgido en el año 2000, hace uso del protocolo HTTP para el envío de mensajes, y puede utilizar lenguajes como XML o JSON.

### SOAP

Este protocolo utiliza XML como lenguaje de codificación, y su principal ventaja es que es agnóstico respecto a la capa de transporte, es decir, que puede ser transmitido a través de HTTP, TCP/IP, SMTP o cualquier otro, a diferencia de REST que únicamente opera sobre HTTP/HTTPS.



## REST

Su filosofía se basa en la ausencia de estado y la “equivalencia” entre los cuatro verbos que pueden utilizarse en el protocolo HTTP y las cuatro operaciones CRUD (Create, Read, Update, Delete) básicas que pueden realizarse sobre una fuente de datos. Es el protocolo más utilizado en servicios web abiertos o públicos.

GET ↔ SELECT (Obtener)

POST ↔ INSERT (Crear)

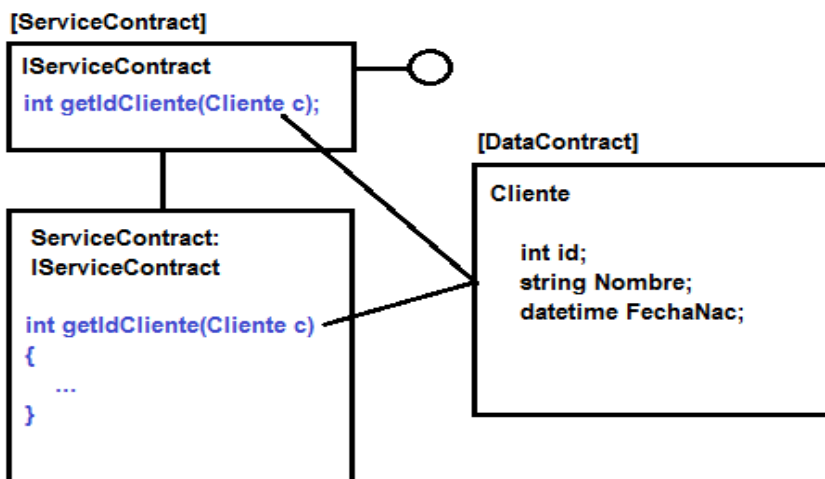
PUT ↔ UPDATE (Actualizar)

DELETE ↔ DELETE (Borrar)

## ¿Cómo logra WCF esta independencia?

Básicamente, mediante la separación entre operaciones y datos. Al igual que hacemos uso de interfaces en nuestro código para aislar la firma de los métodos de sus implementaciones, un servicio web WCF utiliza este mismo concepto: establece un contrato a través de una interfaz (que será adornado con atributos específicos) que una clase se encargará de implementar. De este modo, un servicio WCF, se compondrá, a grandes rasgos, de:

- **Contrato de servicio (ServiceContract):** expone una operación que nuestro servicio web es capaz de ejecutar. Corresponde a una interfaz.
- **Contrato de datos (DataContract):** implementa un tipo de dato que el servicio web será capaz de manejar. Generalmente, será el tipo de dato que manejará el contrato de servicio.
- **Implementación del servicio:** implementará la interfaz correspondiente al contrato de servicio, haciendo uso del contrato de datos para intercambiar la información.





Con esto podemos entender el éxito de WCF en la plataforma .NET: permite olvidarse del transporte y centrarse en la codificación de la lógica del servicio, ahorrando una enorme cantidad de tiempo y recursos y simplificando enormemente la interoperabilidad, ya que, si en un determinado momento es necesario un protocolo determinado para comunicarse con un cliente preexistente, basta con configurar un endpoint para que ese cliente pueda acceder a nuestro servicio en lugar de codificar toda la lógica de comunicaciones que implicaría un servicio web convencional.

### REFERENCIAS

[http://profesores.elo.utfsm.cl/~agv/elo330/2s05/projects/CesarVasquez/sitio\\_web/que\\_es.html](http://profesores.elo.utfsm.cl/~agv/elo330/2s05/projects/CesarVasquez/sitio_web/que_es.html)  
<https://www.techopedia.com/definition/16394/windows-communication-foundation-wcf>  
<https://danielggarcia.wordpress.com/2013/12/11/wcf-i-introduccion-a-windows-communication-foundation/>