

¿Como usar Java RMI?

Para utilizar Java RMI se debe seguir los siguientes pasos:

1. Para cada objeto remoto se debe crear una **interface I** que defina el prototipo de cada método a exportar. Es necesario declarar que los métodos remotos pueden producir la excepción *java.rmi.RemoteException*. La **interface I** debe **heredar** de *java.rmi.Remote*.

2. El *código de los métodos* remotos se debe **escribir** en una **clase C** que **implemente** la **interface I**. La **clase C** debe ser una *subclase* de *java.rmi.server.UnicastRemoteObject*. El **constructor** default de la **clase C** debe **invocar** el *constructor de la superclase*. Es necesario declarar que los métodos remotos pueden producir la excepción *java.rmi.RemoteException*.

3. El *proceso servidor* deberá **registrar** la **clase C** invocando el **método bind()** o el **método rebind()** de la clase *java.rmi.Naming*. A los métodos **bind()** y **rebind()** se les pasa como **parámetros** la **URL** correspondiente al *objeto remoto* y una instancia de la **clase C**. La URL tiene la siguiente forma: *rmi://ip.puerto/nombre*, donde *ip* es la dirección IP de la computadora dónde ejecuta el programa *rmiregistry*, *puerto* es el número de puerto utilizado por *rmiregistry* (se puede omitir si *rmiregistry* utiliza el puerto default 1099) y *nombre* es el nombre con el que identificaremos el objeto.

4. El *proceso cliente* deber **invocar** el método **lookup()** de la clase *java.rmi.Naming* para **obtener** una *referencia al objeto remoto*. El método **lookup()** **regresa** una *instancia de la clase Remote*, la cual se debe **convertir** al tipo de la **interface I** mediante casting. Utilizando la referencia, *el proceso cliente* **invocará** los métodos remotos de la **clase C**.

Por razones de seguridad, la aplicación *rmiregistry* se debe ejecutar en la misma computadora dónde ejecuta el servidor.

Por default la aplicación *rmiregistry* utiliza el puerto 1099, si se utiliza otro puerto, se deberá pasar el número de puerto como argumento al ejecutar *rmiregistry*.

Se puede notar que el proceso servidor permanece en ejecución debido a que los métodos **bind()** y **rebind()** crean threads que no terminan.