



UNIVERSIDAD **CATÓLICA**
de Colombia

Remote Method Invocation

Diego Alberto Rincón Yáñez MSc

darincon@ucatolica.edu.co

Sistemas Distribuidos

Departamento de Ingeniería de Sistemas y Computación

Afiliada a la Federación Internacional de Universidades Católicas (FIUC)

w w w . u c a t o l i c a . e d u . c o



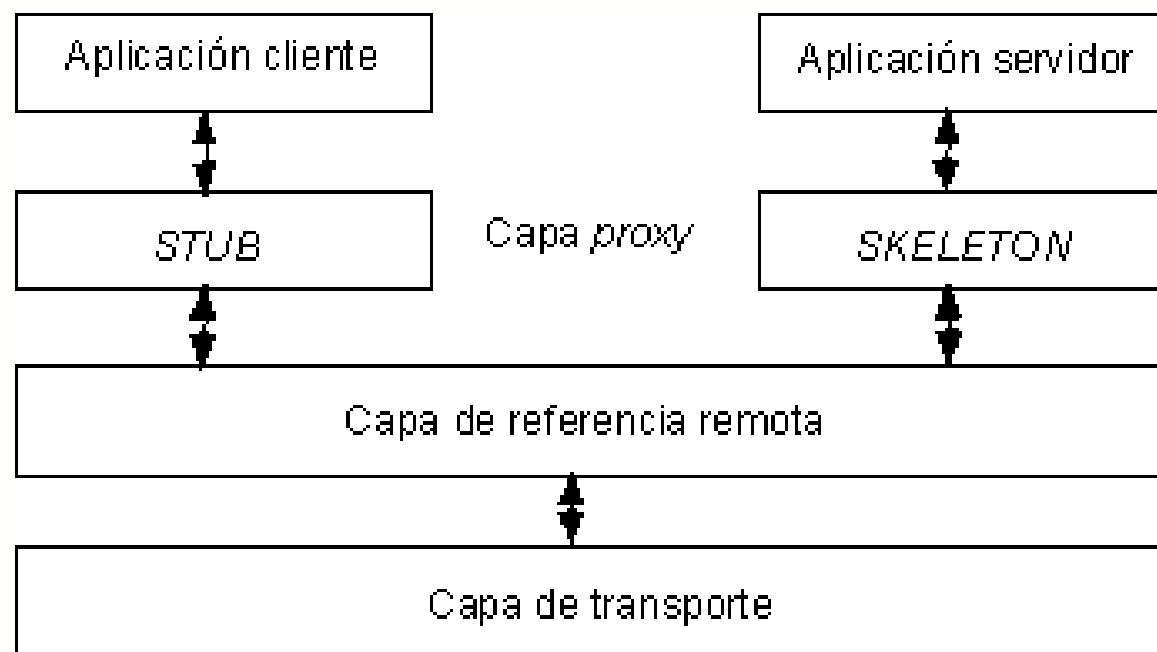
RMI (*Remote Method Invocation*)

Es un mecanismo que permite realizar llamadas a métodos de objetos remotos situados en distintas (o la misma) máquinas virtuales de Java, compartiendo así recursos y carga de procesamiento a través de varios sistemas.



RMI (*Remote Method Invocation*)

- La arquitectura RMI puede verse como un modelo de cuatro capas:





RMI (*Remote Method Invocation*)

- La primera capa es la de **aplicación** y se corresponde con la implementación real de las aplicaciones cliente y servidor. Aquí tienen lugar las llamadas a alto nivel para acceder y exportar objetos remotos. Cualquier aplicación que quiera que sus métodos estén disponibles para su acceso por clientes remotos debe declarar dichos métodos en una interfaz que extienda `java.rmi.Remote`.



RMI (*Remote Method Invocation*)

- La capa 2 es la capa **proxy**, o capa stub-skeleton. Esta capa es la que interactúa directamente con la capa de aplicación. Todas las llamadas a objetos remotos y acciones junto con sus parámetros y retorno de objetos tienen lugar en esta capa.



RMI (*Remote Method Invocation*)

- La capa 3 es la de **referencia remota**, y es responsable del manejo de la parte semántica de las invocaciones remotas. También es responsable de la gestión de la replicación de objetos y realización de tareas específicas de la implementación con los objetos remotos, como el establecimiento de las persistencias semánticas y estrategias adecuadas para la recuperación de conexiones perdidas.



RMI (*Remote Method Invocation*)

- La capa 4 es la de transporte. Es la responsable de realizar las conexiones necesarias y manejo del transporte de los datos de una máquina a otra. El protocolo de transporte subyacente para RMI es JRMP (*Java Remote Method Protocol*), que solamente es “comprendido” por programas Java.



RMI (*Remote Method Invocation*)

- Toda aplicación RMI normalmente se descompone en 2 partes:
 - Un **servidor**, que crea algunos objetos remotos, crea referencias para hacerlos accesibles, y espera a que el cliente los invoque.
 - Un **cliente**, que obtiene una referencia a objetos remotos en el servidor, y los invoca.