

01/03/2024

# Desarrollo De Aplicaciones Distribuidas

**Investigación:**

RMI Desarrollo de sistemas distribuidos

**Integrantes:**

Perea Valdez Leonardo Alberto



1. La invocación remota de métodos en Java es un modelo de objetos distribuidos diseñado específicamente para ese lenguaje, manteniendo la semántica de su modelo de objetos locales. Esto facilita la implantación y el uso de objetos distribuidos.
2. En el modelo de objetos distribuidos de Java, un objeto remoto es aquel cuyos métodos pueden ser invocados por objetos que se encuentran en una máquina virtual (MV) diferente.
3. RMI (Invocación Remota de Métodos) en Java proporciona funciones para localizar objetos remotos, comunicarse con ellos, y cargar el código de operación que implementa las clases que son pasadas por valor.
4. La facilidad de configuración y la integración con Java son características destacadas de RMI, lo que lo hace una herramienta útil para el desarrollo de sistemas distribuidos.
5. La seguridad en RMI es un aspecto importante a considerar, especialmente en el contexto de atravesar firewalls, donde la capa de transporte de RMI incluye la llamada remota dentro del protocolo HTTP.
6. El documento también menciona la comparación de RMI con otros modelos de objetos distribuidos como CORBA y DCOM, resaltando las características y niveles de abstracción de cada uno.
7. El proceso de desarrollo de aplicaciones con RMI incluye la definición de la interfaz remota, la programación de la clase implementación, la compilación de la clase implementación, la ejecución del compilador de stubs, y el arranque del registro RMI en el servidor.
8. El paso de parámetros en RMI tiene una semántica específica, donde los tipos primitivos se pasan por valor, la referencia a un objeto distribuido se pasa por referencia, y los objetos normales serializables se pasan por valor.
9. La escalabilidad, interoperabilidad y estándar abierto son aspectos relevantes a considerar al trabajar con RMI, ya que influyen en la capacidad de las aplicaciones para crecer, interactuar con otros sistemas y seguir estándares abiertos.
10. El documento también incluye ejemplos prácticos de implementación de RMI, como un chat que puede recibir múltiples usuarios a la vez para enviarse mensajes unos a otros.

1. "La invocación remota de métodos de Java es un modelo de objetos distribuidos, diseñado específicamente para ese lenguaje, por lo que mantiene la semántica de su modelo de objetos locales, facilitando de esta manera la implantación y el uso de objetos distribuidos." - Este punto se refiere a cómo el modelo de objetos distribuidos en Java permite la invocación de métodos en objetos que se encuentran en máquinas virtuales diferentes, manteniendo la semántica de los objetos locales.

Ejemplo breve: Un objeto en una máquina virtual puede invocar métodos en un objeto remoto que se encuentra en otra máquina virtual, como si estuvieran en la misma máquina.

2. "Los sistemas distribuidos requieren que las partes que los componen y que se ejecutan en diferentes espacios de direcciones, tengan la capacidad de comunicarse entre sí." - Este punto destaca la necesidad de comunicación entre las partes de un sistema distribuido que se ejecutan en diferentes espacios de direcciones.

Ejemplo breve: En un sistema distribuido, un servidor que se ejecuta en una máquina debe ser capaz de comunicarse con múltiples clientes que se ejecutan en diferentes máquinas.

3. "La principal desventaja de los objetos distribuidos de Java, con respecto a las llamadas a procedimientos remotos y Sockets, es definitivamente el rendimiento." - Este punto señala que la principal desventaja de los objetos distribuidos de Java es su rendimiento en comparación con las llamadas a procedimientos remotos y sockets.

Ejemplo breve: Las llamadas a procedimientos remotos y sockets pueden tener un rendimiento más rápido que los objetos distribuidos de Java en ciertos escenarios.

El punto central que considero como un punto clave para una lluvia de ideas es la comparación entre CORBA, DCOM y RMI en el contexto de sistemas distribuidos. Esta comparación es crucial para comprender las diferentes tecnologías disponibles y sus respectivas ventajas y desventajas en el desarrollo de aplicaciones distribuidas.

CORBA (Common Object Request Broker Architecture), DCOM (Distributed Component Object Model) y RMI (Remote Method Invocation) son tecnologías utilizadas para la comunicación entre objetos distribuidos. Cada una tiene sus propias características y niveles de abstracción, lo que las hace adecuadas para diferentes escenarios de desarrollo.

CORBA es conocida por su interoperabilidad y su capacidad para comunicarse con objetos escritos en diferentes lenguajes de programación. DCOM, por otro lado, es una tecnología de Microsoft que se centra en la comunicación entre componentes distribuidos en entornos Windows. Por último, RMI es específico de Java y está diseñado para soportar llamadas a procedimientos remotos entre objetos que se ejecutan en Máquinas Virtuales Java (JVM).