

Nombre de la práctica: Operaciones como servicios en una arquitectura cliente/servidor heterogénea

1. Objetivo

Los alumnos aplicarán el concepto de RPC y Java RMI para la comunicación de procesos en una arquitectura de tipo cliente/servidor replicada.

2. Descripción

Los alumnos (equipo) deben proponer un proyecto para aplicar los conceptos de RPC y Java RMI describiendo el problema, los objetivos, metas y la propuesta de solución para el problema planteado. Deberán diseñar e implementar dicha solución como un conjunto con, al menos, 9 servicios ofrecidos por un servidor y, al menos, un cliente que realice la invocación de los servicios. Los nueve servicios deberán estar agrupados en 3 categorías. Por ejemplo: salud, finanzas, videojuegos, etc. Las categorías deberán estar implementadas en: una en RPC, una en Java RMI y la tercera es libre (RPC o RMI).

3. Desarrollo del servidor

El servidor implementará un conjunto de 9 procedimientos y métodos ofrecidos como servicios, agrupados en 3 categorías. Los procedimientos deberán permitir la comunicación de la siguiente manera: una categoría en RPC, una en Java RMI y la tercera es libre (RPC o RMI).

Todos los servicios, tanto procedimientos RPC como métodos Java RMI, deben tener parámetro de envío, y retornar un valor, según sea el caso. No se permite procedimientos ni métodos con valor de retorno *void*.

4. Replicación de un conjunto de servicios.

Se debe implementar proponer un escenario de replicación parcial, es decir, replicando solo un conjunto de servicios, es decir una de las tres categorías: RPC, RMI o libres (RPC o RMI).

5. Desarrollo del cliente

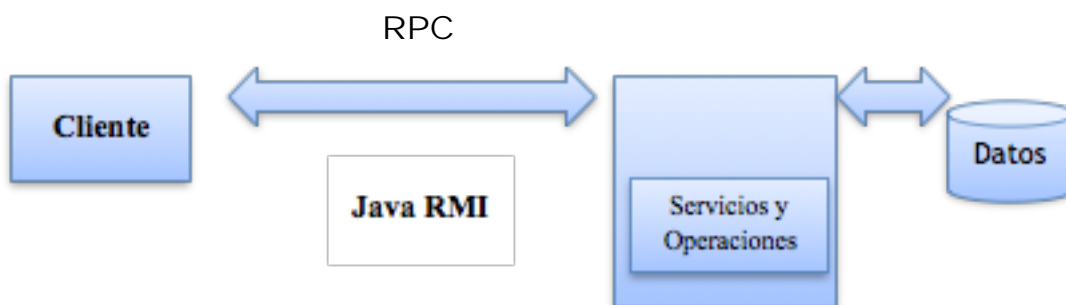
El cliente será una aplicación de consola, Web, de escritorio o móvil que deberá permitir el llamado a cualquier tipo de servicio ofrecido por el servidor.

Así pues la aplicación cliente deberá proporcionar un menú donde el usuario podrá, elegir el servicio y luego ingresar o proporcionar el o los parámetros correspondientes.

Por ultimo, el cliente deberá recibir el resultado del servidor y mostrar el resultado en pantalla de una manera amigable (tablas, texto, explicación, etc.).

6. Recomendaciones generales

La aplicación cliente y servidor deberán estar codificadas en Java. El uso de base de datos, archivos o cualquier medio de almacenamiento queda libre. La arquitectura de la aplicación se muestra a continuación:



7. Entrega

- Primer conjunto de documentos impresos, utilizando las plantillas de entrega proporcionadas, una para cada inciso, que contenga: a) descripción del problema y objetivos; b) propuesta de solución con diseño del proyecto: diagrama de componentes detallados y la descripción de cada componente. **Lunes 18 de noviembre de 2019 a las 5:30 p.m. en el aula E-306.**
- Segundo conjunto de documentos impresos, utilizando las plantillas de entrega proporcionadas, una para cada inciso, que contenga: a) desarrollo de cada componente de la arquitectura, puede apoyarse de fragmentos de código; b) pruebas con figuras (capturas de pantallas) para cada una, y/o tablas de resultados: **viernes 22 de noviembre de 2019 a las 5:30 p.m. en el aula E-306.**
- Revisiones del funcionamiento general, trabajo en equipo e individual: **viernes 22 de noviembre de 2019 a las 5:30 p.m. en el aula E-306.**
- Entrega del código fuente de la aplicación, tanto el cliente como el servidor: enviar el código fuente a más tardar el miércoles 22 de noviembre de 2019 a las 6:00 p.m., a los correos: jaro@correo.azc.uam.mx y con copia para: carlos.alberto.mm@hotmail.com

Nota: Se revisará el funcionamiento del programa de manera individual y en equipo.

8. Evaluación

- Implementación y funcionamiento correcto de XML-RPC y Java RMI: 40 %
- Implementación y funcionamiento correcto de Replicación: 20 %
- Evaluación individual y trabajo en equipo: 20 %
- Documentación del proyecto: 20 %