

# Tarea 6. Modelo RPC

---

## SISTEMAS CONCURRENTES Y DISTRIBUIDOS

Ignacio David Vázquez Pérez

218292866

---

### Objetivo:

Conocer los pormenores de uno de los modelos más utilizados en los sistemas distribuidos, el modelo RPC (llamada a procedimiento remoto).

### Introducción:

El Modelo de Llamada a Procedimiento Remoto, comúnmente conocido como RPC (por sus siglas en inglés, Remote Procedure Call), es un paradigma fundamental en la programación y comunicación de sistemas distribuidos. Proporciona una forma efectiva de invocar funciones o procedimientos en un sistema remoto como si fueran locales, lo que facilita la comunicación y la colaboración entre procesos que se ejecutan en diferentes nodos de una red. Esta tarea proporcionará una comprensión sólida del Modelo RPC y su relevancia en el contexto de la programación y comunicación en sistemas distribuidos.

### Actividades a realizar

#### Clases de fallas en los sistemas RPC

1. El cliente no puede localizar al servidor, ¿que sucede, que problemáticas se enfrentan, se puede solucionar?
  - **Problema de Resolución de Nombres:** Uno de los problemas más comunes es la incapacidad del cliente para resolver el nombre o la dirección del servidor. Esto puede deberse a una configuración incorrecta de la red, un servidor fuera de servicio o un nombre de servidor incorrecto.
    - **Solución:** Para abordar este problema, se deben verificar las configuraciones de resolución de nombres en el cliente y el servidor. Asegurarse de que se utilice el nombre o la dirección IP correctos del servidor y de que los servicios de resolución de nombres, como DNS (Domain Name System) o archivos hosts, estén configurados adecuadamente.
  - **Firewalls y Reglas de Seguridad:** En algunos casos, los cortafuegos o reglas de seguridad en la red pueden bloquear la comunicación entre el cliente y el servidor, impidiendo que el cliente localice al servidor.
    -

- **Solución:** Es importante revisar y ajustar las reglas de seguridad del cortafuegos o de otros dispositivos de seguridad de la red para permitir la comunicación entre el cliente y el servidor en los puertos y protocolos necesarios.
- **Servidor Fuera de Servicio:** Si el servidor está temporalmente fuera de servicio o experimenta una interrupción, el cliente no podrá localizarlo.
  - 
  - **Solución:** En este caso, la solución puede ser esperar a que el servidor vuelva a estar en línea o tomar medidas para redundancia, como tener servidores de respaldo disponibles para garantizar la disponibilidad continua de los servicios.
- **Problemas de Conexión:** Los problemas de conexión, como enlaces de red interrumpidos o caídas en la conexión, pueden hacer que el cliente no pueda localizar al servidor.
  - **Solución:** Para abordar problemas de conexión, es importante verificar la integridad de la red y solucionar cualquier problema con los enlaces o la conectividad. Esto puede requerir la intervención del equipo de soporte de red.
  - **Errores en la Configuración del Cliente:** A veces, los errores en la configuración del cliente, como direcciones IP incorrectas o configuraciones de puerto, pueden evitar que el cliente se comuniquen con el servidor.
  - **Solución:** Revise y verifique la configuración del cliente para asegurarse de que la información de dirección y puerto sea precisa y coincida con la configuración del servidor.

2. El cliente no puede localizar al servidor, ¿qué sucede, qué problemáticas se enfrentan, se puede solucionar?

- **Problemáticas:**
  - **Fallo en la Entrega de la Solicitud:** El problema principal es que la solicitud del cliente no llega al servidor, lo que significa que el servidor no tiene conocimiento de la solicitud y, por lo tanto, no puede responder a ella. Esto puede deberse a diversas razones, como problemas de red, pérdida de paquetes, congestión de la red o errores en la capa de transporte.
  - **Tiempo de Espera del Cliente:** El cliente puede quedar en un estado de espera indefinido esperando una respuesta del servidor que nunca llegará. Esto puede llevar a bloqueos o retrasos en la operación del cliente y afectar negativamente la experiencia del usuario.
  - **Posible Pérdida de Datos:** Si la solicitud del cliente contiene datos importantes o transacciones críticas, la pérdida de la solicitud puede resultar en la falta de actualización de datos o pérdida de información importante.
- **Soluciones:**
  - **Reintento de la Solicitud:** Una solución común es implementar mecanismos de reintento en el cliente. Cuando se pierde una solicitud, el cliente puede intentar nuevamente enviarla después de un breve período de tiempo. Esto puede ayudar a superar problemas temporales en la red o en el servidor.

- **Confirmación de Entrega:** Se puede implementar un sistema de confirmación de entrega para asegurarse de que la solicitud haya llegado al servidor. El servidor puede enviar una confirmación al cliente una vez que haya recibido la solicitud, y el cliente puede retransmitir la solicitud si no recibe la confirmación dentro de un tiempo razonable.
- **Implementar Retransmisiones Confiables:** En protocolos de comunicación confiables, como TCP/IP, se incluyen mecanismos de retransmisión de datos para garantizar que los paquetes lleguen al destino de manera confiable. Si se utiliza un protocolo confiable, la retransmisión se gestionará automáticamente.
- **Detección de Pérdida de Conexión:** El cliente puede implementar mecanismos para detectar la pérdida de la conexión con el servidor y, en caso de detectarla, intentar restablecer la conexión o notificar al usuario sobre el problema.
- **Registro de Errores:** Es importante registrar y monitorear errores de comunicación para identificar problemas recurrentes y tomar medidas para resolverlos.
- **Optimización de la Red:** Para abordar problemas de congestión de red o pérdida de paquetes, se pueden realizar mejoras en la infraestructura de red, como la optimización de enlaces, la implementación de redundancia de rutas o la reducción de la carga de la red.

3. Se pierde el mensaje de respuesta del servidor al cliente. ¿Qué significa que una solicitud sea idempotente?

En el contexto de protocolos de comunicación y operaciones en un sistema distribuido, una "solicitud idempotente" se refiere a una solicitud que puede repetirse varias veces sin causar efectos secundarios adicionales más allá del resultado de la primera ejecución. En otras palabras, realizar la misma solicitud varias veces no produce un cambio adicional en el estado del sistema más allá del primer intento. Esto es especialmente relevante en situaciones en las que la solicitud o su respuesta pueden perderse o duplicarse debido a problemas de comunicación.

4. ¿que sucede, que problemáticas se enfrentan, se puede solucionar?

Cuando se pierde el mensaje de respuesta del servidor al cliente, se presentan diversas implicaciones y problemáticas que pueden afectar la comunicación y la integridad de los datos entre el cliente y el servidor. Sin embargo, es importante aclarar que el término "solicitud idempotente" se refiere a un concepto diferente en el contexto de protocolos de comunicación y no está directamente relacionado con la pérdida de mensajes de respuesta.

- **Problemáticas:**

- **Falta de Confirmación:** La pérdida del mensaje de respuesta implica que el cliente no recibe confirmación de que su solicitud fue procesada correctamente. Esto puede generar incertidumbre en el cliente acerca del estado de la operación y si se completó con éxito.
- **Reintentos:** El cliente puede enfrentar la decisión de si debe o no volver a enviar la misma solicitud debido a la falta de respuesta. Esto podría resultar en la duplicación de acciones no deseadas si la solicitud original ya se procesó en el servidor.

- **Integridad de los Datos:** En aplicaciones que involucran transferencia de datos importantes, como transacciones financieras, la pérdida de respuestas puede comprometer la integridad de los datos y la consistencia de la información entre el cliente y el servidor.

- **Posibles Soluciones:**

- Implementar un mecanismo de confirmación y reintento similar a las soluciones mencionadas en la pérdida de la solicitud del cliente al servidor. El servidor podría enviar confirmaciones de recibido y procesamiento, y el cliente podría retransmitir la solicitud si no recibe confirmación.

## 5. El servidor falla antes de recibir una solicitud

¿que sucede, que problemáticas se enfrentan, se puede solucionar?

- **Qué Sucede:**

- **Solicitud No Atendida:** La solicitud enviada por el cliente al servidor no puede ser procesada, ya que el servidor ha fallado antes de recibir y procesar la solicitud. El servidor no tiene conocimiento de la solicitud del cliente.

- **Problemáticas:**

- **Falta de Servicio:** La principal problemática es que el cliente no puede acceder a los servicios proporcionados por el servidor, ya que el servidor está inactivo debido a la falla. Esto puede afectar la disponibilidad de la aplicación o sistema distribuido.
- **Incertidumbre:** El cliente puede quedar en un estado de incertidumbre sobre si la solicitud se envió con éxito o no, ya que no recibe ninguna confirmación o respuesta del servidor.

- **Posibles Soluciones:**

- **Detección de Fallos:** Para abordar este problema, se pueden implementar mecanismos de detección de fallos tanto en el cliente como en el servidor. El cliente podría intentar detectar la indisponibilidad del servidor mediante mecanismos como el tiempo de espera (timeout) o el uso de señales de salud del servidor. Si el cliente detecta la falla del servidor, puede tomar medidas, como reintentar la solicitud más tarde o redirigir la solicitud a un servidor de respaldo si está disponible.
- **Respuesta Automática del Servidor de Respuesta:** En algunos casos, se pueden utilizar servidores de respaldo que estén diseñados para responder automáticamente a las solicitudes del cliente en caso de que el servidor principal falle. Esto implica la replicación de servicios o la implementación de redundancia en la infraestructura del servidor.
- **Colas de Mensajes:** Los sistemas pueden implementar colas de mensajes para almacenar temporalmente las solicitudes del cliente antes de que se procesen. Si el servidor falla antes de recibir una solicitud, la solicitud quedará en la cola de mensajes y se procesará cuando el servidor vuelva a estar disponible.

## 6. El cliente falla después de enviar una solicitud.

¿que sucede, que problemáticas se enfrentan, se puede solucionar?

- **Qué Sucede:**

- **Solicitud No Procesada:** El cliente envía una solicitud al servidor, pero luego experimenta una falla antes de recibir la respuesta del servidor. El servidor ya ha recibido la solicitud, pero el cliente no está en condiciones de procesar la respuesta.

- **Problemáticas:**

1.

1. **Incertidumbre:** El servidor no tiene conocimiento de la falla del cliente y continúa procesando la solicitud, generando una respuesta que el cliente nunca recibirá. Por lo tanto, el cliente queda en un estado de incertidumbre sobre si la solicitud se procesó correctamente o no.
2. **Posible Pérdida de Recursos:** Dependiendo de la naturaleza de la solicitud y la falla del cliente, pueden ocurrir situaciones en las que los recursos del servidor se utilicen para procesar una solicitud que nunca se utilizará. Esto puede llevar al desperdicio de recursos.

- **Posibles Soluciones:**

- **Implementación de Tiempo de Espera (Timeout):** Para abordar esta problemática, el cliente puede implementar un mecanismo de tiempo de espera (timeout). El cliente establece un límite de tiempo durante el cual espera la respuesta del servidor. Si la respuesta no se recibe dentro de ese período, el cliente considera la solicitud como fallida y puede tomar medidas, como registrar el error o reintentar la solicitud.
- **Manejo de Excepciones:** El cliente puede implementar un manejo de excepciones robusto para tratar los casos en los que una solicitud no se procesa correctamente debido a una falla del cliente. Esto puede incluir la captura de excepciones y la implementación de Colas de Mensajes: Los sistemas pueden implementar colas de mensajes para almacenar temporalmente las solicitudes del cliente antes de que se procesen. Si el servidor falla antes de recibir una solicitud, la solicitud quedará en la cola de mensajes y se procesará cuando el servidor vuelva a estar disponible. Colas de Mensajes: Los sistemas pueden implementar colas de mensajes para almacenar temporalmente las solicitudes del cliente antes de que se procesen. Si el servidor falla antes de recibir una solicitud, la solicitud quedará en la cola de mensajes y se procesará cuando el servidor vuelva a estar disponible. Implementación de lógica para revertir cualquier acción que se haya realizado antes de la falla.
- **Mensajes de Estado y Confirmaciones:** El servidor puede enviar mensajes de estado al cliente para informarle sobre el progreso de la solicitud o su estado final. El cliente debe estar diseñado para manejar estos mensajes de estado y confirmaciones, lo que puede ayudar a mitigar la incertidumbre.
- **Implementación de Peticiones Idempotentes:** Si es posible, el diseño de las solicitudes y servicios RPC debería considerar la idempotencia. Una solicitud idempotente es aquella que, si se envía múltiples veces, produce el mismo resultado que si se enviara solo una vez. Esto puede ayudar a mitigar los efectos adversos de las fallas del cliente en algunas situaciones.

7. Cuando un procedimiento no puede encontrar su destino se genera un huérfano, ¿Qué se puede hacer con los huérfanos? Explicalo detalladamente.

Cuando un procedimiento no puede encontrar su destino y se genera un "huérfano" en el contexto de los sistemas de comunicación, se deben tomar medidas para manejar adecuadamente esta situación. Los huérfanos son conexiones o solicitudes que no pueden completarse debido a que el servidor destino no está disponible o no se puede encontrar.

- **Exterminación:**

- La exterminación implica descartar o eliminar la solicitud o conexión huérfana de manera inmediata. Esta acción se toma cuando se determina que la solicitud no es recuperable o que la operación ya no tiene sentido debido a que el servidor destino no puede ser localizado.

- **Reencarnación:**

- La reencarnación implica volver a intentar la solicitud huérfana en un momento posterior. El cliente puede programar automáticamente un nuevo intento de comunicación con el servidor destino después de un período de espera definido. Este enfoque es útil cuando se espera que el servidor esté temporalmente fuera de línea o indisponible.

- **Reencarnación Sutil:**

- Similar a la reencarnación, la reencarnación sutil implica reintentar la solicitud, pero con una estrategia más cuidadosa y adaptativa. En lugar de realizar intentos de manera repetitiva y constante, se pueden aplicar estrategias de espera y retroceso exponencial para evitar una sobrecarga en el servidor.

- **Expiración:**

- La expiración significa que se establece un límite de tiempo para la solicitud huérfana. Si el servidor destino no se encuentra antes de que expire el tiempo especificado, la solicitud se marca como caducada y se descarta. Esto evita que la solicitud quede en espera indefinidamente.

- **RPC (Remote Procedure Call):**

- El protocolo RPC permite que un programa solicite un servicio o función que se ejecuta en un servidor remoto como si fuera una llamada local a una función. Los parámetros se transmiten entre el cliente y el servidor, y la respuesta se envía de vuelta al cliente. Se utiliza en la comunicación entre procesos y sistemas distribuidos.

- **RMI (Remote Method Invocation):**

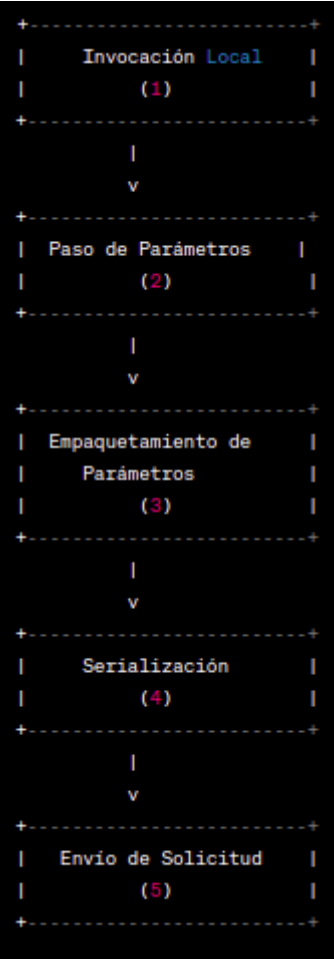
- RMI es un protocolo utilizado en sistemas Java que permite que los objetos Java se comuniquen a través de una red. Permite que los métodos de objetos en una máquina virtual Java (cliente) se invoquen en objetos en otra máquina virtual Java (servidor). Es una forma de RPC específica para aplicaciones Java.

8. Además, a la serie de instrucciones que se ejecutan con cada RPC se llama la ruta crítica. Investiga cual es la ruta crítica y los 14 pasos de la RPC desde el cliente hasta al servidor, describe cada parte y dibuja los dos diagramas correspondientes a cada tema.

La "ruta crítica" en el contexto de las RPC (Remote Procedure Calls) se refiere al conjunto de instrucciones o pasos que son esenciales para la ejecución exitosa de una llamada de procedimiento remoto. Estos pasos representan el flujo crítico de la comunicación entre el cliente y el servidor, asegurando que la solicitud se realice de manera efectiva y que los resultados se devuelvan adecuadamente.

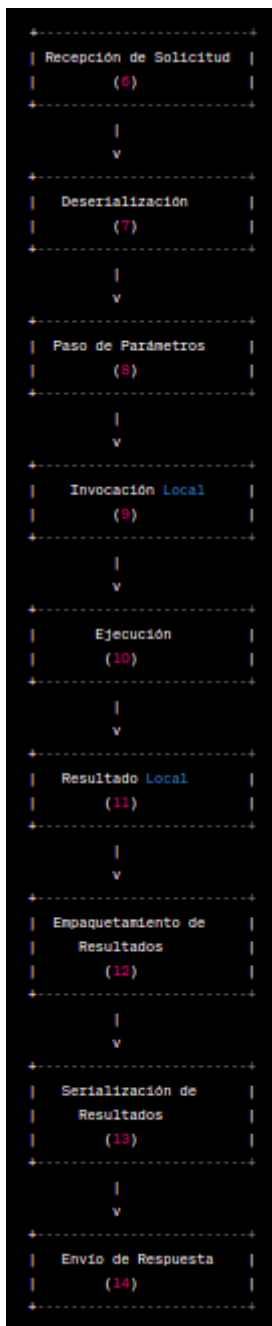
1. **Invocación Local:** El proceso cliente invoca una función o método localmente como si estuviera llamando a una función normal.
2. **Paso de Parámetros:** Los parámetros de la función se pasan al stub (proxy del cliente) localmente.
3. **Empaquetamiento de Parámetros:** El stub empaqueta los parámetros en un mensaje de solicitud.
4. **Serialización:** Los parámetros se serializan en formato de bytes para que puedan transmitirse a través de la red.
5. **Envío de Solicitud:** El mensaje de solicitud se envía desde el cliente al servidor a través de la red.
6. **Recepción de Solicitud:** El servidor recibe el mensaje de solicitud.
7. **Deserialización:** Los parámetros se deserializan en el servidor para que puedan ser utilizados como datos locales.
8. **Paso de Parámetros:** Los parámetros se pasan desde el stub del servidor al procedimiento o método local.
9. **Invocación Local:** El procedimiento o método local en el servidor se invoca usando los parámetros recibidos.
10. **Ejecución:** La lógica del procedimiento o método se ejecuta en el servidor.
11. **Resultado Local:** El resultado de la ejecución se obtiene en el servidor.
12. **Empaquetamiento de Resultados:** El stub del servidor empaqueta el resultado en un mensaje de respuesta.
13. **Serialización de Resultados:** El resultado se serializa en bytes para su transmisión.
14. **Envío de Respuesta:** El mensaje de respuesta se envía desde el servidor al cliente.

**Cliente:**



**Servidor**





## Conclusión

En resumen, la solución a la falla del cliente para localizar al servidor depende de la causa subyacente, que puede variar desde problemas de resolución de nombres hasta problemas de conectividad. Identificar la causa raíz, verificar la configuración y planificar para la alta disponibilidad son pasos clave para restaurar la comunicación entre el cliente y el servidor.

La pérdida de mensajes de solicitud del cliente al servidor es un problema significativo que puede abordarse mediante mecanismos de reintento, confirmación de entrega y monitoreo y optimización de la infraestructura de red.

Una solicitud idempotente es aquella que no tiene efectos secundarios adicionales al repetirse. Las colas de mensajes pueden utilizarse para gestionar solicitudes pendientes en caso de fallas del servidor.

Los diagramas de la "ruta crítica" en una RPC ilustran visualmente los pasos esenciales en la comunicación eficiente entre el cliente y el servidor. Comprender estos pasos es fundamental para el diseño y la depuración de sistemas distribuidos confiables y eficientes.

## Bibliografía

- Tanenbaum Andrew. (1995). Sistemas Operativos Distribuidos. España. Prentice-Hall Hisp.
- McIver Ann. (2011). Sistemas Operativos. México. Cengage Learning.
- Tanenbaum, A., & Van Steen M. (2008). Sistemas Distribuidos, Principios y Paradigmas. (Segunda ed.). Prentice Hall.
- Tanenbaum, A. (2011). Redes de Computadoras. (Quinta ed.). Prentice Hall.
- Elmasri, R., Gil Carrick, A., & Levine, D. (2010). Sistemas Operativos, Un enfoque en espiral. McGraw-Hill.
- IBM. (s.f.). Llamada a procedimiento remoto. <https://www.ibm.com/docs/es/aix/7.3?topic=concepts-remote-procedure-call>
- Santos, L. (2020). Guía gRPC: Introducción. <https://blog.lsanatos.dev/guia-grpc-1/>
- Microsoft. (s.f.). Modelo RPC de Microsoft. <https://learn.microsoft.com/es-es/windows/win32/rpc/microsoft-rpc-model>
- IONOS. (s.f.). Remote procedure call (RPC): ¿en qué consiste esta tecnología? <https://www.ionos.es/digitalguide/servidores/know-how/que-es-rpc/>