



**SCD.**

## **Tarea 5: Modelo cliente-servidor**

**Padilla Perez Jorge Daray**

**02/09/2024**

**CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS**

**Prof. Gutiérrez Salmerón Martha del Carmen**

## Contenido

Introducción.....	3
Contenido.....	4
Modelo cliente-servidor.....	<b>Error! Bookmark not defined.</b>
Modelo “Requerimiento-Respuesta” sin Conexión:.....	<b>Error! Bookmark not defined.</b>
Conclusión.....	4
Referencias.....	15

## Introducción

La comunicación cliente-servidor es un concepto fundamental en el ámbito de las redes y sistemas distribuidos. Se refiere a la interacción entre dos entidades: el cliente (que solicita servicios o recursos) y el servidor (que proporciona esos servicios o recursos). A través de protocolos y primitivas, esta comunicación permite que aplicaciones, sitios web y servicios se conecten y compartan información de manera eficiente.

## Contenido

### Clases de fallas en los sistemas RPC

El cliente no puede localizar al servidor.

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

Causas de que el Cliente no Pueda Localizar al Servidor:

**Problemas de Resolución DNS:** Si el cliente no puede resolver el nombre del servidor a una dirección IP, no podrá establecer una conexión.

**Servidor Inactivo o No Disponible:** Si el servidor está apagado, no responde o no está en línea, el cliente no podrá encontrarlo.

**Errores de Configuración:** Configuraciones incorrectas en el cliente o el servidor pueden dificultar la localización.

**Bloqueo por Firewall o Filtro de Red:** Si hay restricciones de red, el cliente podría estar bloqueado para acceder al servidor.

Problemáticas Asociadas:

**Tiempo de Espera Excesivo:** El cliente puede pasar mucho tiempo esperando una respuesta del servidor.

**Interrupción del Servicio:** Si el cliente no puede localizar al servidor, los servicios que dependen de él no estarán disponibles.

**Frustración del Usuario:** Los usuarios pueden sentirse frustrados si no pueden acceder a los recursos que necesitan.

Posibles Soluciones:

**Verificar la Conectividad:** Asegurarse de que el cliente pueda comunicarse con el servidor mediante ping u otras herramientas de diagnóstico.

**Revisar Configuraciones:** Verificar las configuraciones de red en ambos extremos (cliente y servidor).

**Resolver Problemas DNS:** Asegurarse de que la resolución DNS funcione correctamente.

**Reiniciar el Servidor:** Si el servidor está inactivo, reiniciarlo podría solucionar el problema.

**Revisar Reglas de Firewall:** Asegurarse de que no haya bloqueos en el firewall.

Se pierde el mensaje de solicitud del cliente al servidor.

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

Causas de la Pérdida del Mensaje de Solicitud:

Problemas de Red o Enrutamiento: Puede haber congestión en la red, errores en los routers o problemas en la ruta de comunicación.

Fallas en el Servidor: Si el servidor está sobrecargado, no responde o está inactivo, no procesará la solicitud.

Errores en el Cliente: El cliente podría no estar enviando correctamente la solicitud o podría estar bloqueado por un firewall.

Problemáticas Asociadas:

Tiempo de Espera Infinito: Si el cliente no recibe una respuesta, podría esperar indefinidamente.

Interrupción del Flujo de Trabajo: Si la solicitud es crucial para una operación, su pérdida puede afectar la funcionalidad completa.

Frustración del Usuario: Los usuarios pueden sentirse frustrados si sus acciones no tienen efecto.

Posibles Soluciones:

Reintentar la Solicitud: El cliente puede reenviar la solicitud después de un tiempo.

Implementar Confirmaciones: El servidor podría enviar una confirmación al cliente cuando recibe una solicitud.

Monitoreo y Diagnóstico: Supervisar la red y los servidores para detectar problemas y resolverlos rápidamente.

Se pierde el mensaje de respuesta del servidor al cliente.

**¿Qué significa que una solicitud sea idempotente?**

Causas de la Pérdida del Mensaje de Respuesta:

Problemas de Red o Enrutamiento: Al igual que con las solicitudes, la pérdida de paquetes puede deberse a problemas en la red o en los routers.

Fallas en el Servidor: Si el servidor no responde o está sobrecargado, no podrá enviar la respuesta al cliente.

Errores en el Cliente: El cliente podría no estar esperando la respuesta correctamente o podría haber cerrado la conexión antes de recibirla.

Problemáticas Asociadas:

Tiempo de Espera Infinito: El cliente podría esperar indefinidamente por una respuesta que nunca llega.

Inconsistencia de Datos: Si el cliente no recibe la respuesta, su estado interno podría quedar desactualizado.

Frustración del Usuario: Los usuarios pueden sentirse frustrados si no obtienen la información que esperaban.

¿Qué Significa que una Solicitud Sea Idempotente?

En el contexto de las solicitudes HTTP, una operación idempotente es aquella que puede repetirse varias veces sin cambiar el resultado más allá de la primera ejecución.

Ejemplo: Si una solicitud idempotente se repite, el resultado será el mismo que si se hubiera realizado solo una vez. Esto es crucial para garantizar la integridad y consistencia de los datos en un sistema distribuido.

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

Causas de la Pérdida del Mensaje de Respuesta:

Problemas de Red o Enrutamiento: La pérdida de paquetes puede deberse a congestión en la red, errores en los routers o problemas en la ruta de comunicación.

Fallas en el Servidor: Si el servidor no responde o está sobrecargado, no podrá enviar la respuesta al cliente.

Errores en el Cliente: El cliente podría no estar esperando la respuesta correctamente o podría haber cerrado la conexión antes de recibirla.

Problemáticas Asociadas:

Tiempo de Espera Infinito: El cliente podría esperar indefinidamente por una respuesta que nunca llega.

Inconsistencia de Datos: Si el cliente no recibe la respuesta, su estado interno podría quedar desactualizado.

Frustración del Usuario: Los usuarios pueden sentirse frustrados si no obtienen la información que esperaban.

Posibles Soluciones:

Reintentar la Solicitud: El cliente puede reenviar la solicitud después de un tiempo.

Implementar Confirmaciones: El servidor podría enviar una confirmación al cliente cuando recibe una solicitud.

Monitoreo y Diagnóstico: Supervisar la red y los servidores para detectar problemas y resolverlos rápidamente.

[El servidor falla antes de recibir una solicitud](#)

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

Causas de la Falla del Servidor Antes de Recibir una Solicitud:

**Sobrecarga del Servidor:** Si el servidor está saturado o tiene una alta carga de trabajo, podría colapsar antes de procesar nuevas solicitudes.

**Fallas de Hardware o Software:** Problemas en el hardware (como discos duros defectuosos) o en el software (como errores en el sistema operativo) pueden provocar la caída del servidor.

**Errores de Configuración:** Configuraciones incorrectas pueden afectar la capacidad del servidor para recibir y procesar solicitudes.

**Problemáticas Asociadas:**

**Interrupción del Servicio:** Si el servidor no puede recibir solicitudes, los servicios que ofrece no estarán disponibles.

**Impacto en la Experiencia del Usuario:** Los usuarios experimentarán errores o tiempos de espera prolongados.

**Pérdida de Datos:** Si el servidor falla antes de procesar una solicitud, los datos asociados con esa solicitud podrían perderse.

**Posibles Soluciones:**

**Monitoreo Continuo:** Supervisar el estado del servidor para detectar problemas temprano.

**Balanceo de Carga:** Distribuir las solicitudes entre varios servidores para evitar sobrecargas.

**Respallos y Recuperación:** Implementar sistemas de respaldo y recuperación para minimizar la pérdida de datos.

**El cliente falla después de enviar una solicitud.**

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

**Causas de la Falla del Cliente Después de Enviar una Solicitud:**

**Errores del Cliente:** El cliente podría haber enviado una solicitud incorrecta o incompleta.

**Problemas de Red o Conectividad:** Interrupciones en la red o problemas de conectividad pueden causar que el cliente no reciba la respuesta del servidor.

**Tiempo de Espera Excesivo:** Si el cliente espera demasiado tiempo sin recibir respuesta, podría cerrar la conexión o fallar.

**Problemáticas Asociadas:**

**Transacciones Incompletas:** Si el cliente falla antes de recibir la respuesta, la transacción podría quedar incompleta.

**Inconsistencia de Datos:** Si el cliente no recibe la respuesta, su estado interno podría quedar desactualizado.

Frustración del Usuario: Los usuarios pueden sentirse frustrados si no obtienen la información o el servicio que esperaban.

Posibles Soluciones:

Reintentar la Solicitud: El cliente puede volver a enviar la solicitud.

Implementar Tiempos de Espera Razonables: Configurar tiempos de espera adecuados para que el cliente no cierre la conexión prematuramente.

Monitoreo y Diagnóstico: Supervisar la red y los servidores para detectar problemas y resolverlos rápidamente.

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

### **Solución 1: Exterminación**

La exterminación (o eliminación) de procesos huérfanos es una solución para mitigar los problemas mencionados.

Detección de Procesos Huérfanos:

El sistema operativo debe monitorear continuamente los procesos y detectar aquellos que se han convertido en huérfanos.

Esto se puede hacer mediante el seguimiento de la relación padre-hijo entre procesos.

Terminación Controlada:

Cuando se detecta un proceso huérfano, el sistema operativo puede intentar terminarlo de manera controlada.

Se liberarán los recursos asociados con ese proceso.

Limpieza Regular:

El sistema operativo debe realizar limpiezas regulares para eliminar procesos huérfanos y liberar recursos.

Esto puede ser parte de una rutina de mantenimiento programada.

### **Solución 2: reencarnación**

Concepto:



La reencarnación en sistemas operativos se refiere al proceso de reiniciar o reciclar componentes del sistema para liberar recursos y mejorar la eficiencia.

Es una práctica común en sistemas operativos modernos para mantener un rendimiento óptimo.

Ejemplos de Reencarnación:

Reinicio de Procesos: Los sistemas operativos pueden reiniciar procesos que han estado en ejecución durante mucho tiempo para liberar memoria o recuperar recursos.

Reciclaje de Hilos: Los hilos (subprocesos) pueden ser reutilizados en lugar de crear nuevos cada vez que se necesita procesar una tarea.

Beneficios:

Optimización de Recursos: La reencarnación ayuda a evitar la acumulación innecesaria de procesos o hilos, lo que podría afectar el rendimiento general.

Mejora de la Estabilidad: Al reciclar componentes, se reduce la probabilidad de errores o bloqueos debido a la falta de recursos.

### **Solución 3: reencarnación sutil**

Concepto:

La reencarnación sutil se refiere al proceso de reciclaje o reinicio de componentes del sistema operativo para liberar recursos y mejorar la eficiencia.

Es una práctica común en sistemas operativos modernos para mantener un rendimiento óptimo.

Ejemplos de Reencarnación Sutil:

Reinicio de Procesos: Los sistemas operativos pueden reiniciar procesos que han estado en ejecución durante mucho tiempo para liberar memoria o recuperar recursos.

Reciclaje de Hilos: Los hilos (subprocesos) pueden ser reutilizados en lugar de crear nuevos cada vez que se necesita procesar una tarea.

Beneficios:

Optimización de Recursos: La reencarnación sutil ayuda a evitar la acumulación innecesaria de procesos o hilos, lo que podría afectar el rendimiento general.

Mejora de la Estabilidad: Al reciclar componentes, se reduce la probabilidad de errores o bloqueos debido a la falta de recursos.

## Solución 4: expiración

La técnica de expiración:

En sistemas Unix/Linux, cuando comienza una llamada a procedimiento remoto (RPC), el servidor tiene un cierto tiempo o “quantum” para completar la llamada.

Si el servidor no completa la llamada dentro de ese tiempo, el proceso se considera huérfano y se libera automáticamente.

No se requiere ningún registro específico para esta técnica; simplemente se basa en el tiempo de ejecución.

Ventajas de la expiración:

La expiración garantiza que los procesos huérfanos no permanezcan en el sistema indefinidamente.

Al liberarlos automáticamente después de un período, se evita la acumulación innecesaria de procesos huérfanos.

Además, a la serie de instrucciones que se ejecutan con cada RPC se llama la ruta crítica. Investiga cual es la ruta crítica

### Ruta crítica:

La ruta crítica es un concepto fundamental en la gestión de proyectos. Se utiliza para planificar y controlar eficientemente proyectos complejos.

En la planificación de proyectos, la ruta crítica es la secuencia de actividades que determina la duración total del proyecto.

Cada actividad en la ruta crítica tiene un impacto directo en la fecha de finalización del proyecto. Si alguna de estas actividades se retrasa, todo el proyecto se retrasa.

los 14 pasos de la RPC desde el cliente hasta al servidor.

#### 1. Definición de la interfaz remota:

- En primer lugar, se define una **interfaz remota** que enumera las funciones o procedimientos que el servidor proporcionará.
- Esta interfaz actúa como un **contrato** entre el cliente y el servidor, especificando qué operaciones están disponibles para invocar de forma remota.

#### 2. Implementación de la interfaz en el servidor:

- El siguiente paso es **implementar** las funciones definidas en la interfaz remota en el servidor.
  - Estas funciones son las que el cliente podrá llamar de manera remota.
3. **Inicio del servidor y espera de solicitudes:**
- El servidor se **inicia** y queda a la espera de **solicitudes entrantes**.
  - Cuando llega una solicitud RPC, el servidor está listo para procesarla.
4. **Generación del stub del cliente:**
- El cliente crea un **stub** (o proxy) local para representar la función remota en su espacio de direcciones.
  - El stub actúa como **sustituto** del procedimiento real del servidor.
5. **Empaquetado de parámetros:**
- El stub del cliente recupera los **parámetros necesarios** para la llamada desde el espacio de direcciones del cliente.
  - Luego, convierte estos parámetros en un formato estándar (como Network Data Representation, NDR) para la transmisión a través de la red.
6. **Envío de la solicitud al servidor:**
- El stub del cliente llama a las **funciones de la biblioteca en tiempo de ejecución del cliente RPC** para enviar la solicitud y sus parámetros al servidor.
7. **Transmisión a través de la red:**
- La solicitud RPC y los parámetros se transmiten a través de la red hacia el servidor.
8. **Recepción en el stub del servidor:**
- En el lado del servidor, el **stub del servidor** recibe la solicitud entrante.
9. **Desempaquetado de parámetros:**
- El stub del servidor desempaqueta los parámetros recibidos y los convierte al formato adecuado para su procesamiento local.
10. **Invocación de la función real en el servidor:**
- El stub del servidor llama a la **función real** (implementada en el servidor) utilizando los parámetros desempaquetados.
11. **Ejecución de la función en el servidor:**
- La función real se ejecuta en el servidor, realizando la operación solicitada.
12. **Generación de la respuesta:**

- El resultado de la función se envía de vuelta al **stub del servidor**.

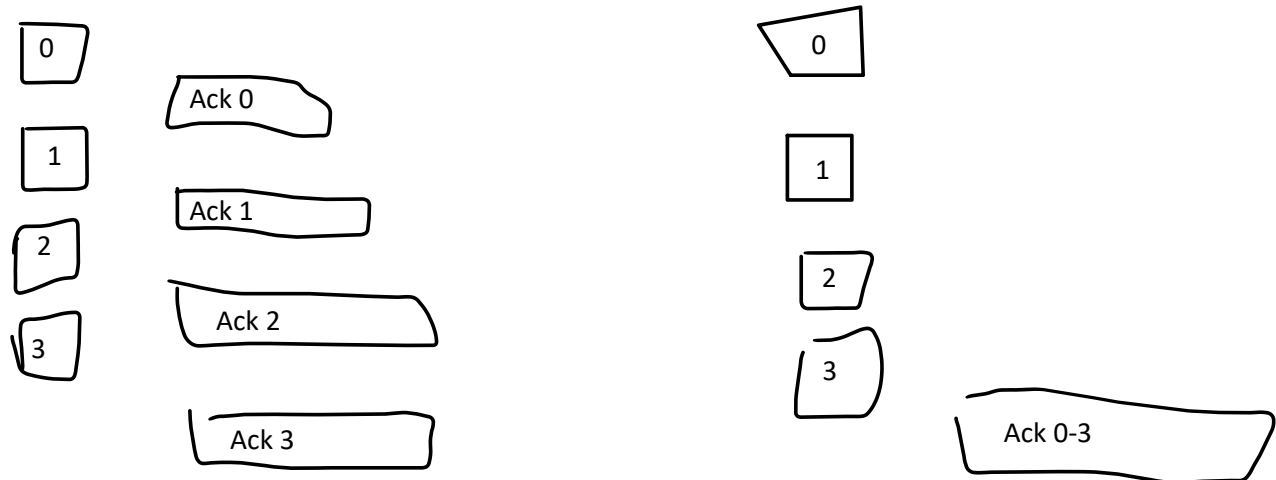
### 13. Retorno al cliente:

- El stub del servidor envía la respuesta al cliente a través de la red.
- El stub del cliente desempaqueta la respuesta y la devuelve al programa cliente.

dibuja los dos diagramas correspondientes a cada tema.

Datos de 4k

0 1 2 3



	Maquina Cliente	Maquina servidor	
Cliente	Llama al procedimiento de resguardo	Realiza el servicio	Servidor
Resguardo del cliente	Prepara el buffer de mensajes Ordena parámetros Llena los campos de encabezado Señala el núcleo	Llama al servidor Establece parámetros en la pila Desordena los parámetros	Resguardo del servidor
Nucleo	Cambio de contexto al núcleo Copia el mensaje del núcleo Determina la dirección de destino Coloca la dirección en el encabezado Inicia el cronometro	Cambio de contexto servidor Copia mensaje servidor Ve si esta esperando Decide a cual resguardado darle Interrupción de proceso	Núcleo

Para finalizar el tema de RPC deberás ver el siguiente video y describe brevemente de lo que trata.

Trata de que es un grupo como se componen, los tipos de grupos que existen, el direccionamiento que el grupo puede tener, el ordenamiento y la escalabilidad de estos.

## Conclusión

En un mundo cada vez más interconectado, la comunicación cliente-servidor sigue siendo esencial. Desde la navegación web hasta las aplicaciones móviles y los servicios en la nube, esta arquitectura subyace en gran parte de nuestra experiencia digital. La confiabilidad, la seguridad y la eficiencia en esta comunicación son aspectos críticos para garantizar una experiencia fluida y satisfactoria para los usuarios.

## Referencias

Tanebaum Andrew. (1995). *Sistemas Operativos Distribuidos*. España. Prentice-Hall Hisp.

Ann, S. (s.f.). *Sistema operativo embebido Definición / explicación:TechEdu*. Obtenido de <https://techlib.net/techedu/sistema-operativo-embebido/>

Antonio, J. (16 de Octubre de 2023). *Sistemas Operativos en Red: Fundamentos y Diferencias Clave:*