



Sistemas Distribuidos

Módulo 2

Comunicación en Sistemas Distribuidos



AGENDA

1. Introducción
 1. Modelos de Comunicaciones.
 2. Tipos de Comunicación.
 3. Paradigmas de Comunicación.
2. Pasaje de Mensajes.
3. Comunicación Directa: mensajes, sockets.
4. Comunicación Remota: request-reply, RPC, RMI.
5. Llamadas a Procedimiento Remoto (RPC): concepto e implementación.
6. Comunicación Indirecta: Grupo, MOM, Publica-Suscribe.
7. Sockets: concepto e implementación.



AGENDA

1. Introducción

1. Modelos de Comunicaciones.
2. Tipos de Comunicación.
3. Paradigmas de comunicación.

2. Pasaje de Mensajes.
3. Comunicación Directa: mensajes, sockets.
4. Comunicación Remota: request-reply, RPC, RMI.
5. Llamadas a Procedimiento Remoto (RPC): concepto e implementación.
6. Comunicación Indirecta: Grupo, MOM, Publica-Suscribe.
7. Sockets: concepto e implementación.

COMUNICACIÓN EN SISTEMAS DISTRIBUIDOS

La comunicación entre procesos necesita compartir información:

a) datos compartidos



b) pasajes de mensajes o copias compartidas





COMUNICACIÓN EN SISTEMAS DISTRIBUIDOS

Tipos de Comunicación

- ✓ Comunicación PERSISTENTE: *almacena* el mensaje (información) enviado por el emisor *el tiempo que tome* entregarlo al receptor.
- ✓ Comunicación TRANSITORIA: *almacena* un mensaje *sólo mientras* las aplicaciones del emisor y receptor están en ejecución.



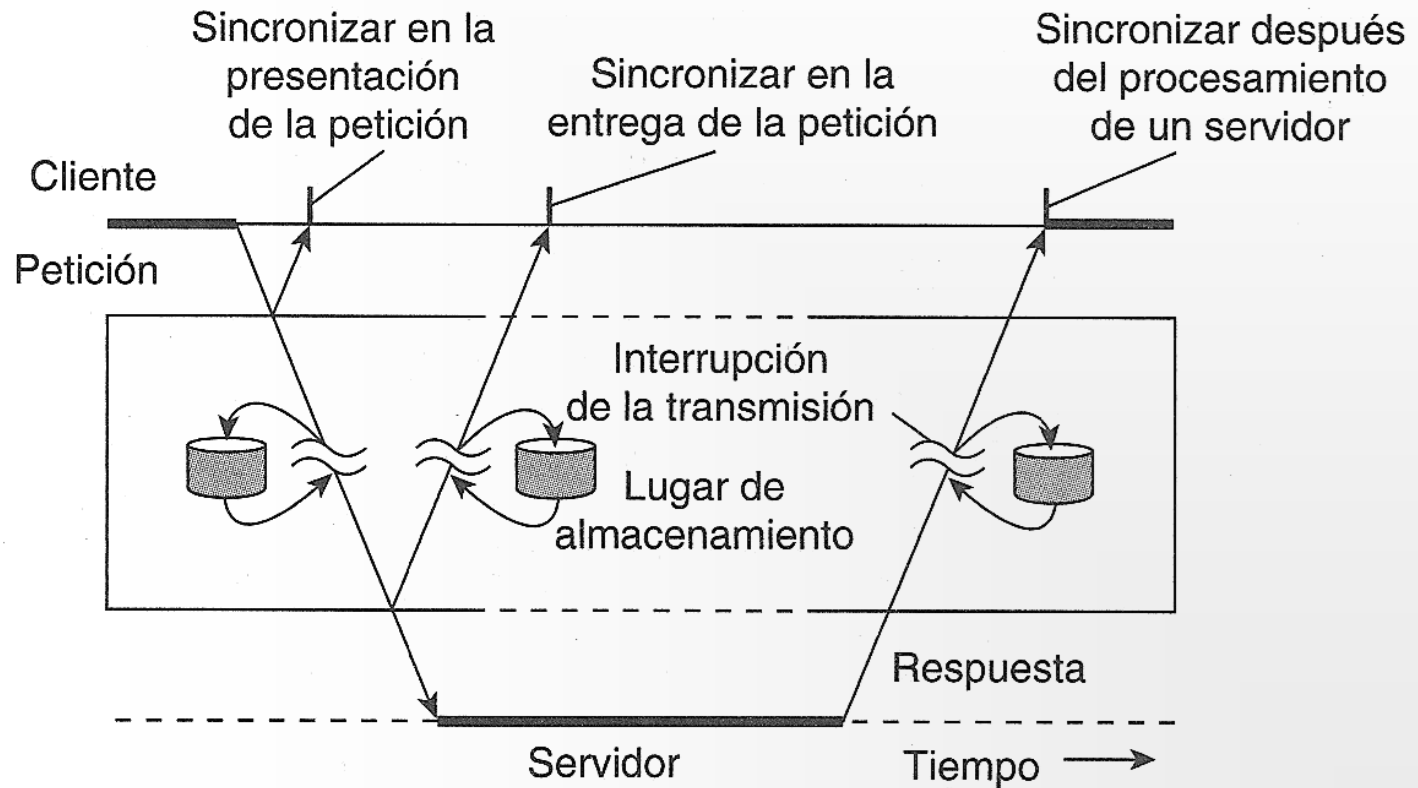
COMUNICACIÓN EN SISTEMAS DISTRIBUIDOS

Tipos de Comunicación

- ✓ Comunicación ASINCRÓNICA: el emisor *continúa inmediatamente* después de que ha pasado su mensaje para la transmisión.
- ✓ Comunicación SINCRÓNICA: el emisor *es bloqueado* hasta que se sabe que su petición es aceptada.

COMUNICACIÓN EN SISTEMAS DISTRIBUIDOS

Ejemplo de Comunicación





COMUNICACIÓN EN SISTEMAS DISTRIBUIDOS

PARADIGMAS DE COMUNICACIÓN

- ✓ Comunicación Directa (entre Procesos)
- ✓ Comunicación Remota
- ✓ Comunicación Indirecta



AGENDA

1. Introducción

1. Modelos de Comunicaciones.
2. Tipos de Comunicación.
3. Paradigmas de comunicación.

2. Pasaje de Mensajes.

3. Comunicación Directa: mensajes, sockets.
4. Comunicación Remota: request-reply, RPC, RMI.
5. Llamadas a Procedimiento Remoto (RPC): concepto e implementación.
6. Comunicación Indirecta: Grupo, MOM, Publica-Suscribe.
7. Sockets: concepto e implementación.

PASAJE DE MENSAJES – CARACTERÍSTICA DESEABLES

SIMPLICIDAD	Simple y fácil de utilizar. Uso directo
SEMÁNTICA UNIFORME	Comunicaciones locales y remotas
FIABILIDAD (CONFIABILIDAD)	Manejo de las fallas
EFICIENCIA	Reducir el número de mensajes intercambiados
FLEXIBILIDAD	Soportar distintos tipos de comunicación
SEGURIDAD	Autenticación del emisor y receptor. Encriptación de los mensajes
PORTABILIDAD	Aplicar a nuevos protocolos. Heterogeneidad
CORRECTITUD	Utilización de multicast. Comunicación de grupos

PASAJE DE MENSAJES – ESTRUCTURA

Una estructura de mensajes típica:

Datos actuales o punteros	Información de estructura		#sec o id del mensaje	Direcciones	
	Número de bytes/elementos	Tipo		recep	env



El *enviador* determina el contenido del mensaje.

El *receptor* tiene en cuenta como interpretar los datos.



PASAJE DE MENSAJES

SINCRONIZACIÓN

No bloqueante

El receptor conoce la llegada del mensaje

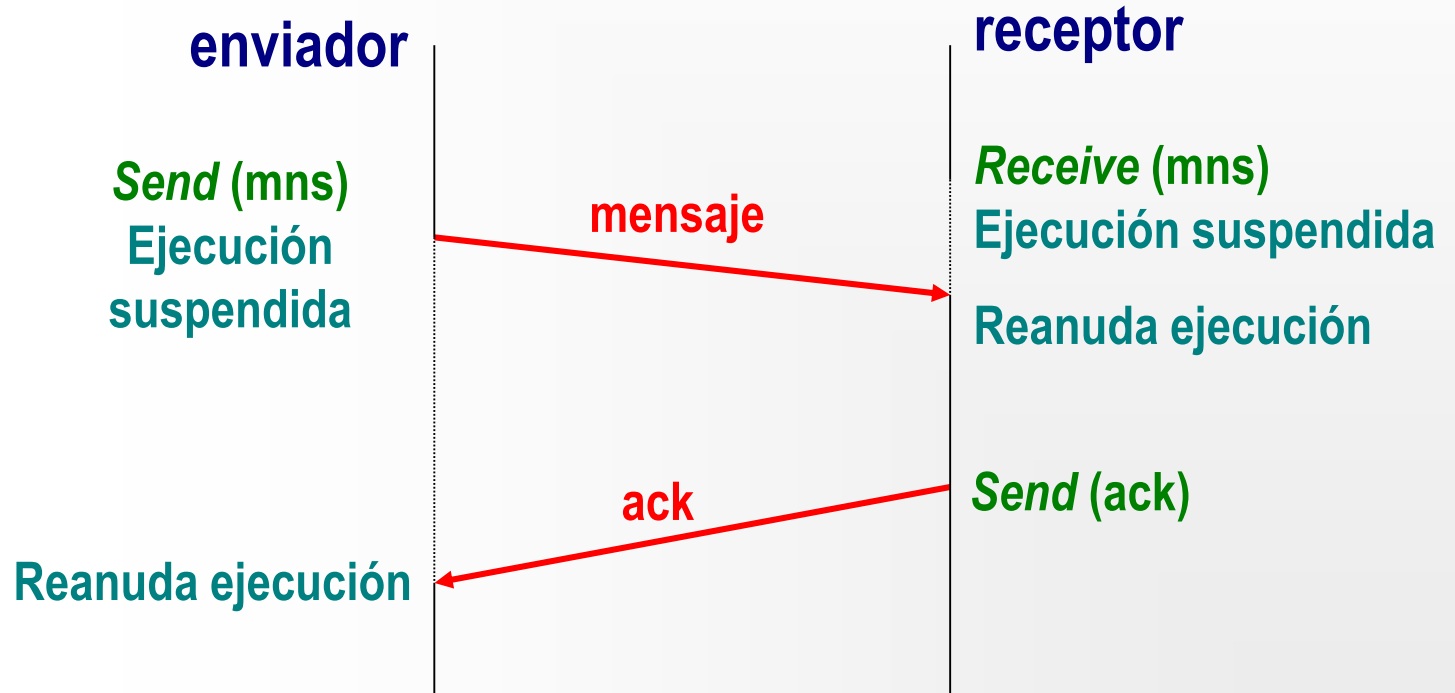
- ✓ Polling
- ✓ Interrupción

Bloqueante \Rightarrow *sincrónica*

Fácil de implementar pero poca concurrencia

PASAJE DE MENSAJES

COMUNICACIÓN SINCRÓNICA - MENSAJES BLOQUEANTES





PASAJE DE MENSAJES

BUFFERING

De *buffer nulo* a *buffer* con capacidad ilimitada

No *buffer*

- ▶ Cita (rendez-vous)
- ▶ Descarte

Buffer simple

Adecuado para transferencia sincrónica

Capacidad infinita

Almacena todo lo que recibe (asincrónica)



PASAJE DE MENSAJES

BUFFERING

***Buffer* límite finito**

Puede haber rebalse de *buffer*

- ✓ Comunicación no exitosa (lo hace menos confiable)
- ✓ Comunicación con flujo controlado (bloquea al enviador hasta que haya espacio)

***Buffer* múltiple**

Mailbox o pórtico



PASAJE DE MENSAJES

MENSAJES MULTIDATAGRAMA

La mayoría tiene un límite superior en el tamaño del dato que puede ser transmitido en algún momento (MTU).

Esto implica que magnitudes mas grandes deben fragmentarse en paquetes.

El ensamblador y desensamblador es responsabilidad del sistema de pasaje de mensajes.



PASAJE DE MENSAJES

CODIFICACIÓN Y DECODIFICACIÓN DE MENSAJES DE DATOS

Un puntero absoluto pierde significado cuando es transmitido de un espacio a otro.

Diferentes programas objeto ocupan una cantidad de espacio variada.

Métodos de Intercambio

- ▶ Formato general acordado
- ▶ Formato emisor



PASAJE DE MENSAJES

CODIFICACIÓN Y DECODIFICACIÓN DE MENSAJES DE DATOS

XDR (External Data Representation)

- ✓ Proceso empaquetado (marshalling)
- ✓ Proceso desempaquetado (unmarshalling)

Se usan, en general, dos representaciones:

- ▶ Representación etiquetada (MACH, XML)
- ▶ Representación no etiquetada (SUN XDR, CORBA CDR)



PASAJE DE MENSAJES

DIRECCIONAMIENTO DE LOS PROCESOS

Direccionamiento Explícito	Direccionamiento Implícito
Problema de nombres de las partes involucradas en una interacción	No se explicita el nombre sino se menciona un servicio
Send (process-id,msg)	Send-any (service-id,msg)
Receive(process-id,msg)	Receive-any (proceso-mudo,msg)



PASAJE DE MENSAJES

MANEJO DE FALLAS

- ▶ Caída de sitio
- ▶ Caída de enlace

Problemas posibles:

- a) Pérdida del mensaje de requerimiento
- b) Pérdida del mensaje de respuesta
- c) Ejecución del requerimiento no exitosa

PROTOCOLOS DE MENSAJES CONFIABLES

Cuatro mensajes

Tres mensajes

Dos mensajes



AGENDA

1. Introducción

1. Modelos de Comunicaciones.
2. Tipos de Comunicación.
3. Paradigmas de comunicación.

2. Pasaje de Mensajes.

3. Comunicación Directa: mensajes, sockets.

4. Comunicación Remota: request-reply, RPC, RMI.

5. Llamadas a Procedimiento Remoto (RPC): concepto e implementación.

6. Comunicación Indirecta: Grupo, MOM, Publica-Suscribe.

7. Sockets: concepto e implementación.



COMUNICACIÓN DIRECTA

- La forma más simple de comunicación entre procesos es a través del pasaje de mensajes.
- El proceso emisor debe especificar el destino del proceso receptor (puede ser con una dirección y un puerto).
- Los mensajes pueden ser:
 - Datagrama
 - Stream
 - Multicast



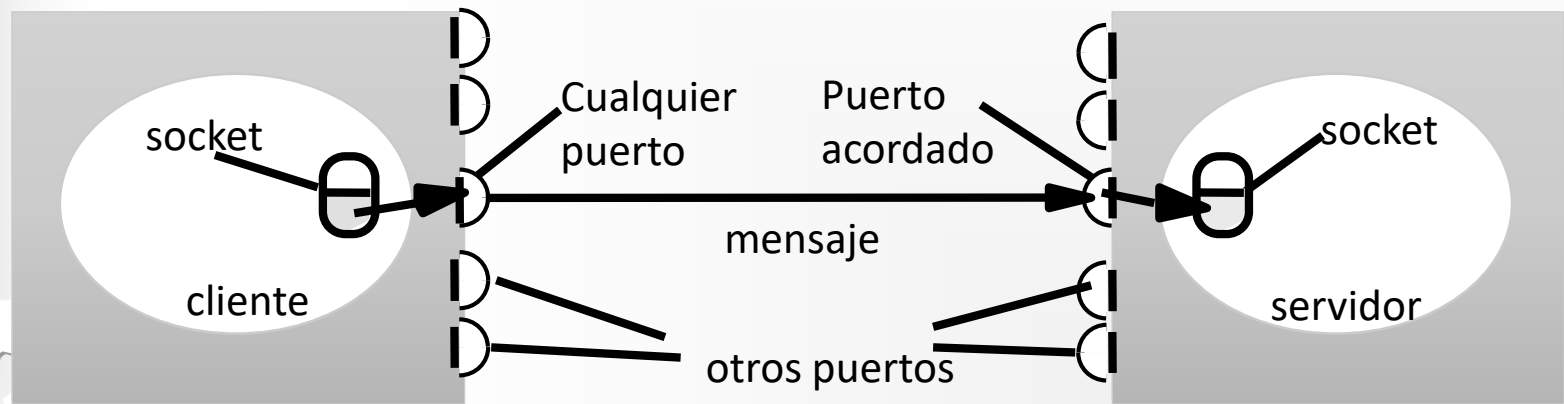
COMUNICACIÓN TRANSITORIA Y ENTRE PROCESOS

Sockets

- ▶ Es una interfaz de entrada-salida de datos que permite la intercomunicación entre procesos.
- ▶ Es un punto final (endpoint) en la comunicación, el cual una aplicación puede escribir datos que serán enviados por la red y desde el cual ingresará los datos que puede leer.

COMUNICACIÓN TRANSITORIA Y ENTRE PROCESOS

- Sockets



Dirección Internet = 138.37.94.248

Dirección Internet = 138.37.88.249



AGENDA

1. Introducción
 1. Modelos de Comunicaciones.
 2. Tipos de Comunicación.
 3. Paradigmas de comunicación.
2. Pasaje de Mensajes.
3. Comunicación Directa: mensajes, sockets.
4. Comunicación Remota: request-reply, RPC, RMI.
5. Llamadas a Procedimiento Remoto (RPC): concepto e implementación.
6. Comunicación Indirecta: Grupo, MOM, Publica-Suscribe.
7. Sockets: concepto e implementación.



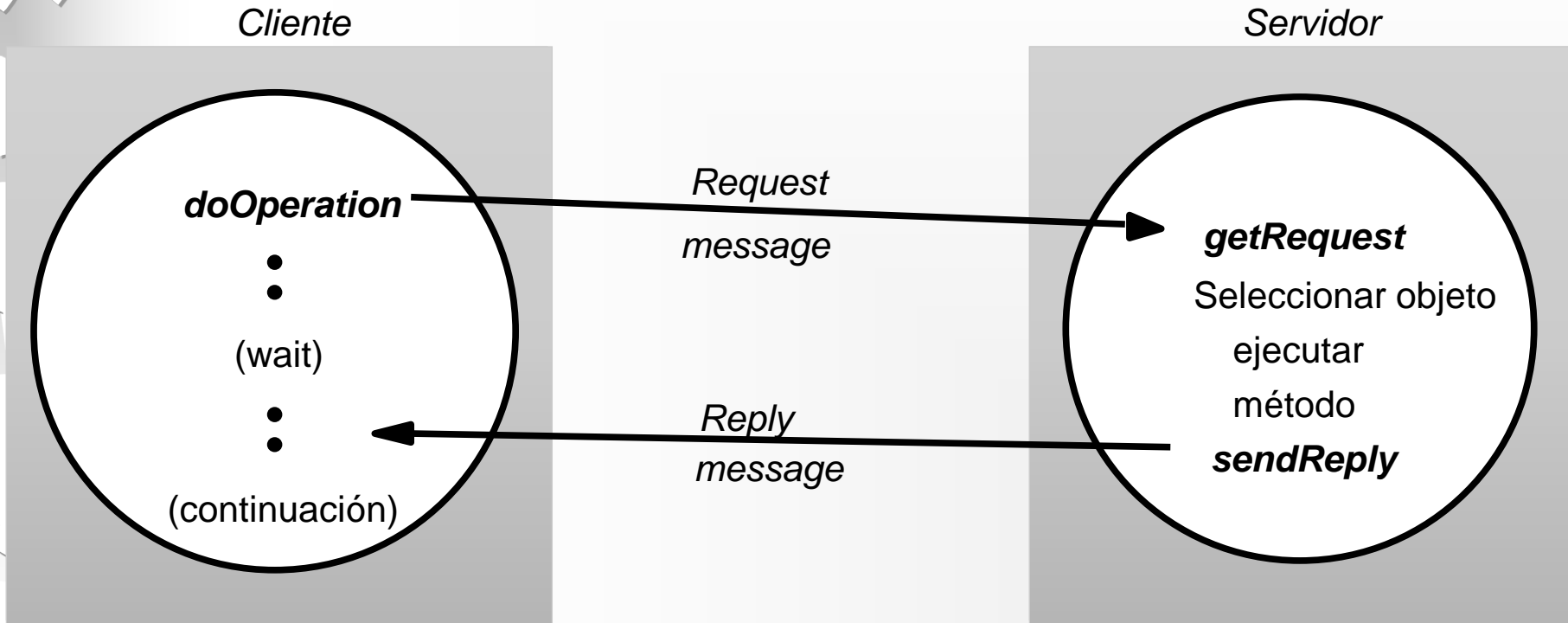
COMUNICACIÓN REMOTA

REQUEST-REPLY

- Soporta el intercambio bidireccional de mensajes.
- Se adapta a los requerimientos del modelo cliente-servidor.
- Es un protocolo de bajo nivel.
- Típicamente la comunicación es de tipo sincrónica.
- Confiable.

COMUNICACIÓN REMOTA

REQUEST-REPLY





Bibliografía:

- Sinha, P. K.; “Distributed Operating Systems: Concepts and Design”, IEEE Press, 1997.
- Tanenbaum, A.S.; van Steen, Maarten; “Distributed Systems: Principles and Paradigms”. 2nd Edition, Prentice Hall, 2007 and 1st Edition 2002.
- Coulouris, G.F.; Dollimore, J. y T. Kindberg; “Distributed Systems: Concepts and Design”. 5th Edition Addison Wesley, 2011.
- van Steen, Maarten; Tanenbaum, A.S; “Distributed Systems”. 3rd Edition, Prentice Hall, 2017.