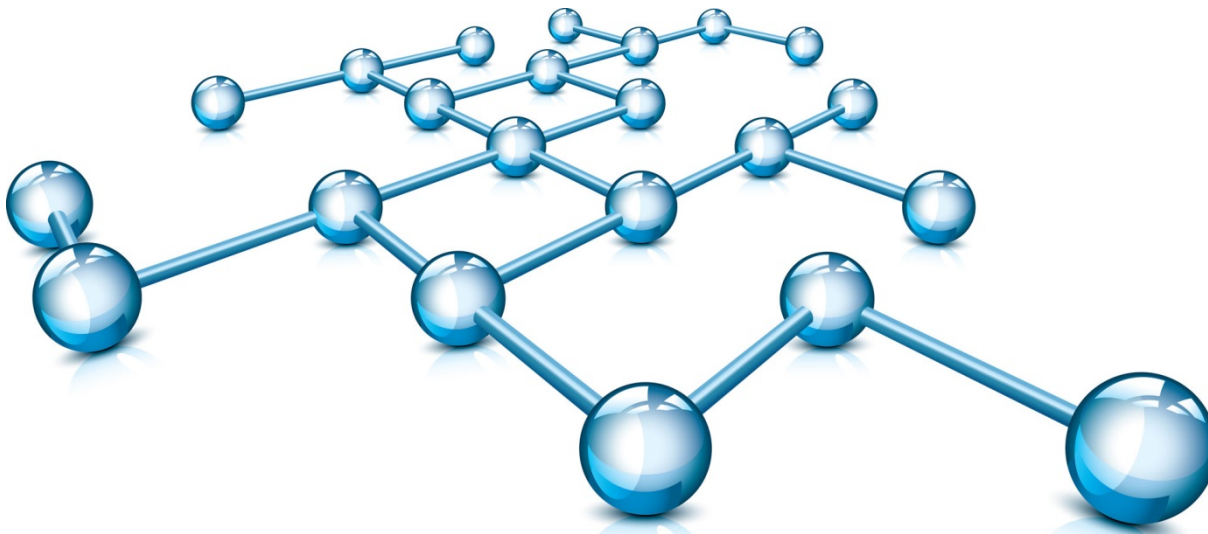


Comunicación entre Procesos

Profesor:

Dr. J. Octavio Gutiérrez García

octavio.gutierrez@itam.mx



Comunicación entre Procesos

Aplicaciones, Servicios

Invocación remota, comunicación indirecta

Primitivas subyacentes de comunicación entre procesos:
Sockets, paso de mensajes, empaquetado y
representación de datos.

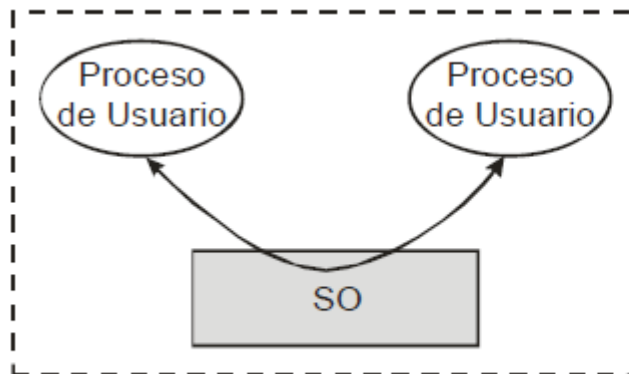
UDP y TCP

Comunicación entre Procesos: Primitivas

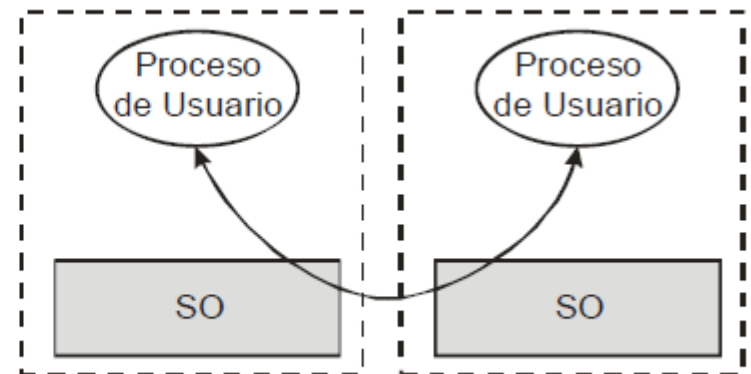


Emisor:
`send(destino, mensaje)`

Receptor:
`receive(origen, mensaje)`



Un computador



Dos computadores

Clasificación de la comunicación

Síncrona

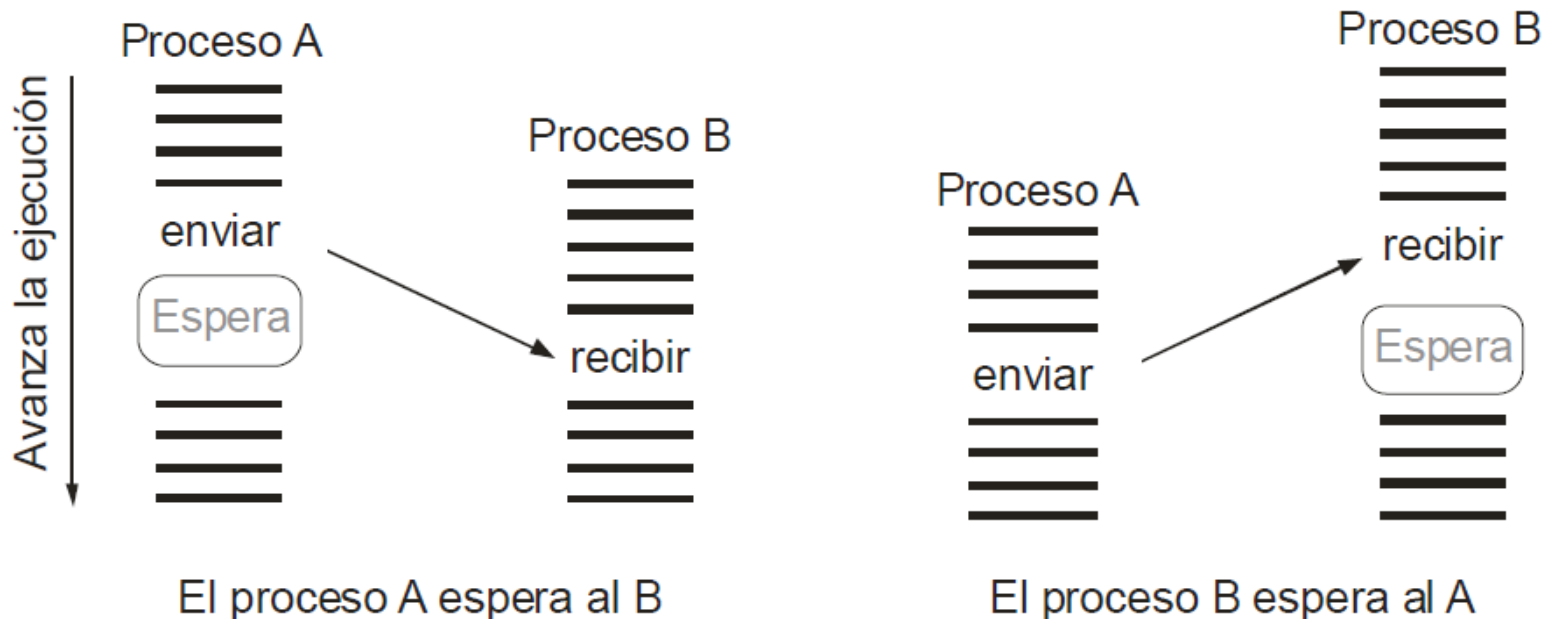


Asíncrona



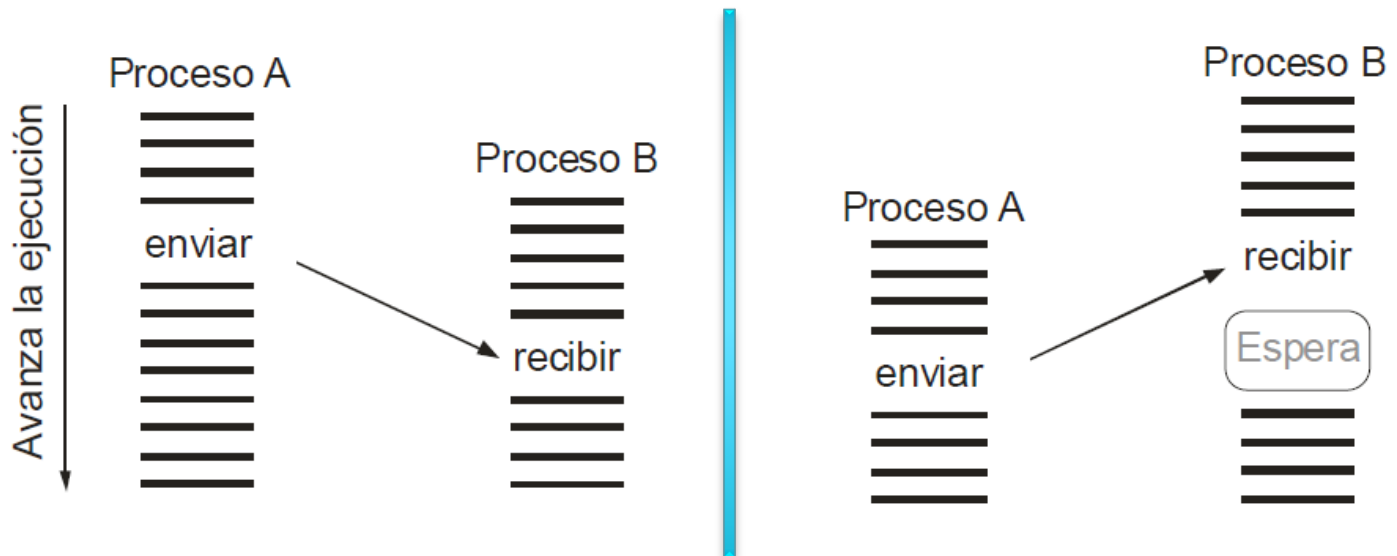
Clasificación de la comunicación

- Síncrona (**bloqueante**)
 - Enviar() y Recibir() son operaciones de bloqueo



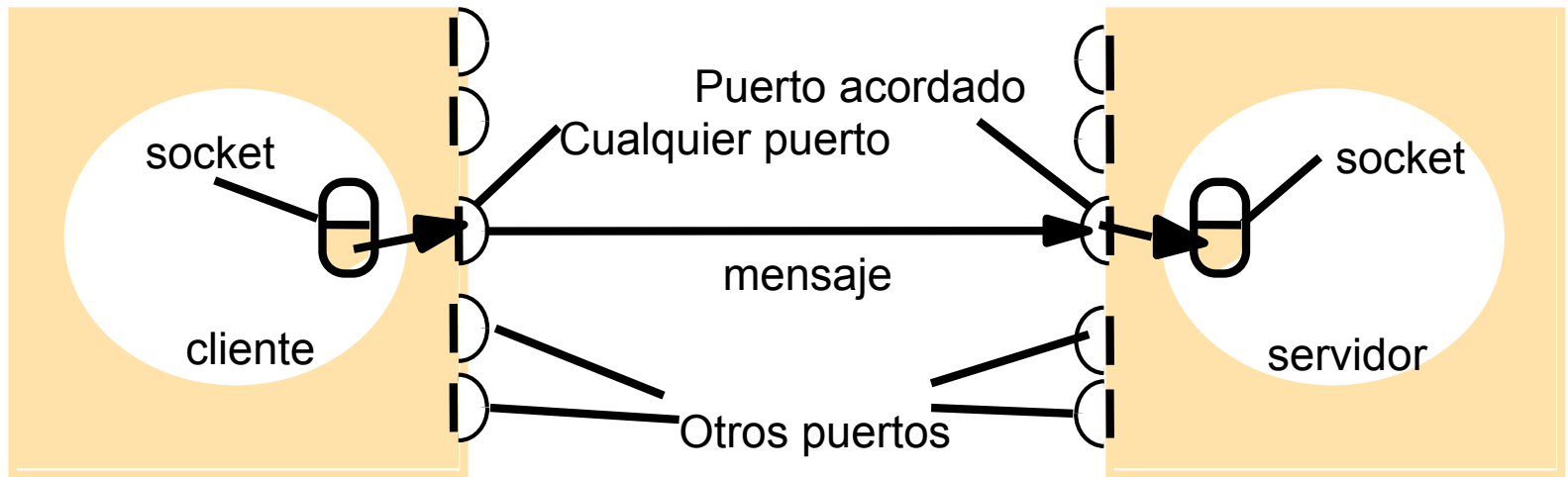
Clasificación de la comunicación

- Asíncrona (**No bloqueante**)
 - Enviar() es no bloqueante
 - Recibir() puede ser bloqueante o no bloqueante.



Sockets

- Tanto UDP como TCP utilizan la abstracción de Sockets.



Dirección IP = 138.37.94.248

Dirección IP = 138.37.88.249

Comunicación de datagramas UDP

- Cuestiones importantes para la comunicación de datagramas usando Sockets:
 - El proceso de recepción debe de especificar un arreglo de bytes (de un tamaño en particular) en el cual se recibirá el mensaje (≤ 8 Kb)

Comunicación UDP en Java

- Librerías

- `java.net.*;`
- `java.io.*;`

- Clases

- `DatagramSocket(serverPort)`
- `DatagramPacket(m, m.length, aHost, serverPort) // envío`
- `DatagramPacket(buffer, buffer.length) // recepción`
- `InetAddress`



Comunicación de datagramas UDP

- Cuestiones importantes para la comunicación de datagramas usando Sockets:
 - Bloqueo
 - Enviar() no bloqueante
 - Recibir() bloqueante – opción otro Hilo.
 - Se puede definir un tiempo límite de espera (Timeouts)

